UNIVERSITÄT
DES
SAARLANDES

CBI CENTER FOR BIOINFORMATICS

SIC Saarland Informatics Campus

Machine Learning Strategies for Drug Sensitivity Prediction and Treatment Optimization in Cancer

Dissertation zur Erlangung des Grades der
Doktorin der Naturwissenschaften (Dr. rer. nat.)
an der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

von

Lea Eckhart

Saarbrücken
2025

Tag des Kolloquiums: 22.10.2025

Dekan der Fakultät: Univ.-Prof. Dr. Roland Speicher

Prüfungsausschuss:

Vorsitzende des Prüfungsausschusses: Prof. Dr. Verena Wolf

Berichterstatter: Prof. Dr. Hans-Peter Lenhof,

Prof. Dr. Volkhard Helms,

Prof. Dr. Oliver Kohlbacher

Wissenschaftlicher Mitarbeiter: Dr. Michael Backenköhler

Acknowledgements

During my time at Saarland University, I had the pleasure of meeting a bunch of different people without whom I would never have started – let alone finished – this thesis:

First, I would like to thank Prof. Dr. Hans-Peter Lenhof for his inspiring ideas, stimulating discussions, critical scrutiny, and words of encouragement throughout the years.

For always creating a pleasant work environment and for many memorable out-of-work experiences, I have to thank Kerstin Lenhof, Nico Gerstner, and Tim Kehl, who accompanied me since the beginning of my studies, as well as the fantastic Nadine Wilhelm. I am also grateful to have met all the curious, ambitious, and talented students and researchers I had the opportunity to work with at Saarland University. I especially appreciate my office neighbors from Volkamer Lab who warmly welcomed me into their vibrant and fun-loving group.

For the exciting and pleasant interdisciplinary collaborations, I would like to thank Prof. Dr. Kerstin Junker and colleagues from the Urology department at Saarland University Medical Center who gave me fascinating insights into the medical perspective of bioinformatics research.

During the good and, especially, the not-so-good times in life, it is invaluable to have friends by your side. I am deeply thankful for ...

... my bioinformatics crew for our awesome cocktail nights and video calls.
... the great people at DFKI Saarbrücken for many cheerful lunch breaks.
... all my friends who have stuck with me through thick and thin, through distance and time.

Last but not least, I have to thank my family, especially my parents, Kathrin Eckhart and Wolf-Dieter Scheid, as well as Ba Thinh Tran. Thank you for always being so open-minded, understanding, encouraging, involved, patient, level-headed, warm-hearted, honest, and loving.

Publications

Parts of this thesis are based on the following collection of journal publications. A complete list of my publications can be found in Appendix A.

Peer-Reviewed Journal Publications

- Lenhof, K., Eckhart, L., Gerstner, N., Kehl, T., & Lenhof, H.-P. (2022). **Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method.** Scientific Reports, 12(1), 13458.
  DOI: 10.1038/s41598-022-17609-x

- Eckhart, L.*, Lenhof, K.*, Rolli, L.-M., Volkamer, A., & Lenhof, H.-P. (2024). **Reliable anti-cancer drug sensitivity prediction and prioritization.**
  Scientific Reports, 14(1), 12303.
  DOI: 10.1038/s41598-024-62956-6

- Eckhart, L., Lenhof, K., Rolli, L.-M., & Lenhof, H.-P. (2024). **A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.**
  Briefings in Bioinformatics, 25(4), bbae242.
  DOI: 10.1093/bib/bbae242

- Lenhof, K., Eckhart, L.*, Rolli, L.-M.*, & Lenhof, H.-P. (2024). **Trust me if you can: a survey on reliability and interpretability of machine learning approaches for drug sensitivity prediction in cancer.**
  Briefings in Bioinformatics, 25(5), bbae379.
  DOI: 10.1093/bib/bbae379

*These authors contributed equally to this work.

- Eckhart, L.*, Rau, S.*, Eckstein, M., Stahl, P. R., Ayoubian, H., Heinzelbecker, J., ... & Junker, K. (2025). **Machine learning accurately predicts muscle-invasion of bladder cancer based on three miRNAs**.
  Journal of Cellular and Molecular Medicine, 29(3), e70361
  DOI: 10.1111/jcmm.70361

Publications in Preparation

- Eckhart, L., Lenhof, K., Herrmann, L., Rolli, L.-M. & Lenhof H.-P. (2025). **How to Predict Effective Drug Combinations - Moving beyond Synergy Scores**.
  Accepted for publication in iScience.
  DOI: 10.1016/j.isci.2025.112622

*These authors contributed equally to this work.

Abstract

The heterogeneity of cancer is a primary challenge for its treatment. Thus, analyzing large multi-omics and drug-screening datasets of cancer cells with machine learning (ML) is promising to study how cellular properties impact drug response and to apply this knowledge for treatment optimization. In this thesis, we used cell line data to build accurate, reliable, and interpretable ML models for personalizing cancer treatment: We conducted the largest benchmarking to date for drug response prediction, investigating various ML and dimension reduction methods. With SAURON-RF, we developed a novel method that, compared to state-of-the-art approaches, strongly improves predictions for drug-sensitive samples, which are particularly relevant for treatment optimization. To enhance model reliability, we built a pipeline that, for the first time, ensures that sensitivity predictions meet user-defined certainty levels for classification and regression. A major goal in treatment optimization is prioritizing treatment options based on their predicted effectiveness. To enable prioritization, we propose a novel sensitivity measure that is comparable across drugs and drug combinations, overcoming the limitations of existing measures. Additionally, we pioneer ML models predicting dose-specific responses to multi-drug therapies for cell lines unseen during model training. Lastly, we developed highly accurate models for predicting muscle invasion in bladder cancer to guide therapy decisions.

Zusammenfassung

Tumor-Heterogenität stellt eine erhebliche Herausforderung für die Krebsbehandlung dar. Die Untersuchung großer Krebszelldatensätze mittels maschinellen Lernens (ML) ist daher vielversprechend, um Zusammenhänge zwischen genetischen Zelleigenschaften und Medikamentenwirksamkeit zu untersuchen und zur Therapieoptimierung zu nutzen. In dieser Arbeit präsentieren wir zelllinienbasierte ML-Modelle zur Personalisierung der Krebsbehandlung: Zunächst haben wir das bisher umfangreichste ML-Benchmarking zur Wirksamkeitsvorhersage von Krebsmedikamenten durchgeführt. Wir haben einen neuartigen ML-Ansatz entwickelt, der Vorhersagen für wirkstoffempfindliche Proben, die hochrelevant für die Therapieoptimierung sind, signifikant verbessert. Zudem haben wir ein Framework implementiert, welches garantiert, dass Klassifikations- und Regressionsmodelle benutzerdefinierte Zuverlässigkeitskriterien erfüllen. Ein Hauptziel personalisierter Medizin ist das Priorisieren von Medikamenten nach ihrer Wirksamkeit. Um die Effizienz verschiedener Medikamente vergleichbar zu machen, schlagen wir ein neues Sensitivitätsmaß vor, das Defizite existierender Maße behebt. Für Kombinationstherapien haben wir Modelle entworfen, die erstmals Dosis-spezifische Wirksamkeitsvorhersagen für Zelllinien ermöglichen, die nicht zum Modelltraining genutzt wurden. Zuletzt haben wir akkurate Modelle zur Vorhersage der Muskelinvasion in Blasentumoren entwickelt, um die Wahl einer geeigneten Behandlung zu unterstützen.

Contents

Chapter 1

Introduction

Despite decades of intensive research [1], cancer remains one of the greatest medical challenges of our time. In 2022 alone, almost 20 million new cancer cases and 10 million cancer-related deaths were reported worldwide [2]. By 2050, the number of new cases is expected to rise to 35 million [2]. Studies also show that many cancer-related deaths could be avoided or delayed through earlier diagnosis, improved treatment, or disease prevention [3, 4].

It is well established that certain lifestyle choices, such as smoking or alcohol consumption, can significantly increase the risk of cancer development [5]. However, some risk factors are more difficult to avoid, e.g., environmental influences such as air and water pollution [6] or ultraviolet sunrays [7]. Additionally, certain cancer predispositions can also be inherited [8] or acquired through aging [9].

To date, the most common options for cancer treatment are surgery, radiation therapy, and systemic treatments such as chemotherapy [10]. While these methods are generally effective, they entail various side effects: Surgeries incur risks such as wound inflammation and anesthetic complications, while radiation may damage healthy cells close to the treatment site. Similarly, most conventional chemotherapeutic drugs are broadly cytotoxic, targeting not only cancer cells but any quickly dividing cells such as hair follicles or cells of the gastrointestinal tract and bone marrow [11]. Side effects of radiation and chemotherapy include fatigue, nausea, and hair loss [12, 13] but also severe organ and nerve damage [14, 15]. Especially in advanced stages of the disease, the last resort is often to find a treatment that will kill the cancer before the treatment kills the patient [16].

As a consequence of these harsh side effects, medical research has aimed to develop treatments that target cancer cells specifically. One example of targeted therapy is the treatment of breast cancer with the drug Trastuzumab: Around 20% to 30% of breast cancers overexpress the HER2 receptor, which is linked to increased tumor proliferation and decreased survival [17]. Trastuzumab attaches to HER2 on the cell surface, thereby interrupting growth signaling cascades and recruiting the immune system to induce apoptosis [17].

A drawback of such targeted therapies is that they are often only effective for a subset of patients [18]. Trastuzumab, for example, requires the overexpression of HER2 and, even under such conditions, only showed response rates of 12% to 34% in clinical trials [17]. A reason for this selective effectiveness of targeted drugs is the inherent heterogeneity of cancer: The specific molecular aberrations that are causal or consequential to cancer development vary not only between patients but also between tumors of the same patient, and even between individual cells of a tumor [19]. The complex interplay of these aberrations and their impact on disease outcome and treatment success is only partially understood to date.

Motivated by such challenges, the area of personalized medicine focuses on tailoring treatments to the individual patient or tumor while minimizing side effects. While the concept of personalized treatment is already centuries old [20], the rise of high-throughput technologies, such as microarrays and RNA sequencing, allows for gathering an unprecedented amount of omics data from cancer cells. This data is extremely valuable not only for identifying personalized treatment strategies and targets but also for furthering our understanding of cancer heterogeneity and for the development of novel, more effective compounds.

To systematically study the connection between molecular properties of cancer cells and their drug response, large amounts of data are required. Due to ethical concerns and challenges like limited sample availability, this data is difficult to obtain from patients directly. Instead, research commonly relies on tumor model systems such as cancer cell lines [21]. Cell lines can be cultivated almost indefinitely, such that they can be easily characterized through various high-throughput techniques, and their response to a multitude of compounds can be tested. Typically, drug response measures such as the IC50 or AUC value are used to quantify the effect of the treatment on cancer cell survival and proliferation [22]. These continuous measures can also be discretized: cases of treatment success, where a drug effectively reduced the amount of live cancer cells, are commonly classified as *sensitive*, whereas ineffective treatments are classified as *resistant*.

Since the resulting datasets are very complex and high-dimensional, a common way to analyze them is machine learning (ML). The typical goal is to perform *drug sensitivity prediction*, where models are designed to predict a continuous or discrete measure of drug response based on a cell line's omics features. Thereby, the models are trained to learn the relationship between cellular properties and drug response with the aim to generate accurate predictions but also to potentially uncover previously unknown

response markers. A plethora of regression and classification approaches for drug sensitivity prediction exists, ranging from simple linear methods to deep neural networks with complex architectures [23–25], often claiming great prediction power.
Still, even though artificial intelligence is omnipresent in many areas of our lives nowadays, there is no widespread use of drug sensitivity prediction models for clinical decision support to date. So, what hinders the translation of such seemingly powerful models into real-world application? What are the shortcomings of the existing models or the data on which they are trained? Might performance claims be exaggerated? And do these models even answer the right questions, or are central issues overlooked?

In this thesis, we address these questions by highlighting various unresolved challenges of ML-based cancer treatment optimization and presenting the approaches we developed to overcome them.

1.1 Contributions and Scope of this Thesis

Whenever artificial intelligence should be used to perform tasks previously left to human experts, skepticism about the trustworthiness of the proposed methods is appropriate, especially in such a sensitive area as healthcare. Since the developed methods should ultimately benefit humans and overcome human limitations and biases, some degree of skepticism is also conductive for promoting the development of better solutions. With this thesis, we address various challenges of trustworthy ML-based treatment optimization in cancer. Our contributions help to overcome several limitations that currently hamper the translation of prediction models into clinical application.
A first crucial step in this direction is identifying requirements that ML models should fulfill to generate trust in their predictions. Hence, the first contribution of this thesis is a **review of trustworthiness in ML-based drug sensitivity prediction** [26]: We identify three core properties of trustworthy ML, namely performance, reliability of individual predictions, and model interpretability: Here, *performance* refers to quantifying the predictive capabilities of a model using error measures such as the mean squared error (regression) or accuracy (classification) that compare the predicted to the actual responses. In contrast, *reliability* describes the degree of trust we have in an individual prediction for a previously unseen sample when no known response is available [27–29]. Lastly, *interpretability* denotes the extent to which a human can understand the decisions of an ML system and the underlying model [30–32]. While performance is routinely evaluated for drug sensitivity prediction approaches, we found that reliability has hardly been addressed. Moreover, while most methods are concerned with model interpretability to some extent, a clear definition of the term was lacking. Thus, our review

proposes a novel taxonomy formalizing different types of interpretability, which refines the definition of interpretability while also highlighting different options for achieving it. With the following contributions, we address different requirements of trustworthy ML-based drug sensitivity prediction. As mentioned above, one straightforward requirement that a trustworthy ML model should fulfill is good prediction performance. Two factors that heavily impact performance are (1) the choice of ML algorithm and (2) the choice of input features: Due to the continuously improving processing power of modern computers, increasingly complex prediction models and algorithms can be developed. Especially deep neural networks are currently highly popular for drug sensitivity prediction [25, 33, 34]. However, the black-box nature of many complex models makes it difficult for humans to understand their decision-making. Additionally, it is rarely fairly evaluated whether the performance of these complex models is truly superior to the performance of simpler, often more interpretable models. Moreover, the cell line characteristics commonly used as model inputs are usually very high-dimensional and can exceed the number of available samples by several orders of magnitude. Such high-dimensional inputs make models prone to overfitting and can also impede their interpretability, underlining the need for appropriate dimensionality reduction (DR). Consequently, our second contribution is a **systematic benchmarking of different ML and DR techniques for drug sensitivity prediction** [35]. With over 16 million trained models, this is the largest drug sensitivity benchmarking to date in terms of the number of investigated DR methods, feature numbers, and hyperparameters. Our results show that relatively simple algorithms like elastic net using relatively few features can significantly outperform more complex models. Most notably, neural networks, even a state-of-the-art deep learning approach, were outperformed in most analyses. We also discuss the inherent differences in interpretability for different ML and DR techniques and showcase how to assess the potential trade-offs between model interpretability and performance.

While performance is a crucial requirement for ML models, what constitutes a good performance is less obvious than it may seem at first glance: Most targeted anti-cancer drugs act highly specific, such that they are only truly effective for a subset of cases [18]. This is reflected in drug response datasets by an underrepresentation of cell lines with a sensitive (as opposed to a resistant) treatment response. This class imbalance was shown to negatively affect the classification performance of ML models for sensitive samples [36–38]. However, the accurate identification of sensitive cell lines is particularly relevant for personalized medicine since they represent cases of presumably effective treatments. Consequently, special attention should be paid to these samples when designing a prediction model. Building on the observation of class imbalance in discrete drug response measures, we showcase that the underrepresentation of sensitive samples also manifests in continuous response measures such as the IC50 value and that this *regression imbalance* likewise affects the regression performance for sensitive samples in various types of ML models. To counteract these imbalances, we developed **SAURON-RF, an ML model based on random forests that can perform classification and regression simultaneously** [39]. SAURON-RF strongly improves both the classification and regression performance for sensitive cell lines compared to conventional ML models, a state-of-the-art random forest approach, and a state-of-the-art deep learning approach. We also show that performing regression and classification simultaneously is superior to performing both tasks separately or sequentially.

Besides performance and interpretability, we identified reliability as a crucial requirement for trustworthy ML in our review discussed above: While ML model development should always strive for accurate predictions, one should never expect a model to be accurate in all cases. Similarly, medical doctors may feel more confident in their assessment of certain cases compared to others. Performance measures calculated on a test set can give an indication of model performance for unseen samples. However, in real-world applications, a known response is usually lacking such that no performance measures can be computed. Consequently, there is a need to estimate the reliability of individual predictions for previously unseen samples. In our fourth contribution, we developed a **framework for reliability estimation** [40] that is based on conformal prediction (CP): Instead of point predictions, CP yields prediction sets (classification) or intervals (regression) that, under defined conditions, are guaranteed to contain the true drug response of a sample at a user-specified certainty level. By applying our CP framework to SAURON-RF, we are the first to achieve certainty guarantees for drug sensitivity prediction in both classification and regression tasks. Furthermore, we demonstrate that CP improves prediction performance by eliminating misclassifications.

To create trust in a model, it should not only perform well, be interpretable and be reliable but it should also be suited for the specific task one wants to solve. Undoubtedly, the accurate prediction of drug responses can be immensely valuable for identifying sensitivity/resistance markers, as well as the development of novel, more effective compounds. However, for the clinical application, a major goal would be drug prioritization: Drug prioritization uses sensitivity prediction to identify a subset of potentially effective drugs for a given patient (cell line) via classification and ranking them by their effectiveness via regression. The realization of drug prioritization was previously hampered by the lack of a sensitivity measure that is comparable across drugs. Thus, our fifth contribution is the development of a **novel drug response measure called CMax viability with across-drug comparability** [40]. It estimates treatment effectiveness when applying a drug's maximum clinically recommended concentration. By combining our novel measure with SAURON-RF and CP, we are the first to perform drug prioritization under user-defined certainty guarantees.

A second aspect besides drug prioritization to tailor models better to real-life circumstances is to enable predictions for treatments with drug combinations: Treating cancer with single drugs, also known as monotherapy, is often less effective compared to administering two or more drugs simultaneously [41]. Furthermore, combination therapy is less susceptible to drug resistance [41], making it a common approach for cancer treatment. Existing approaches for estimating drug combination responses are mainly concerned with predicting synergy scores that measure the synergistic or antagonistic potential of two drugs for a given cell line [42]. However, these scores are based on various assumptions that are unlikely to hold in real life [43–45]. Moreover, a synergetic effect between two drugs does not necessarily guarantee treatment effectiveness [46]. Consequently, we developed ML models that **predict the effect of combination treatments without relying on synergy scores** [47]. Instead, our models predict the inhibition of cell growth after administering specific concentrations of one, two, or more drugs simultaneously. Our method is the first that provides dose-specific predictions for previously unseen cell lines and drugs. Furthermore, any measure of drug response and even synergy scores can be reconstructed from the model predictions, making our approach applicable in various contexts. Lastly, we propose an extension of the CMax viability for two-drug combinations that enables the simultaneous prioritization of mono- and combination therapies.

While the research presented in this thesis focuses mainly on the optimization of drug-based cancer therapy, surgery is still a widely-used means to combat the disease that is commonly combined with other treatment strategies. Our last contribution focuses on the prediction of cancer progression and the related benefit of surgery in individual cases: We apply ML and CP to data from bladder cancer patients to achieve a **reliable prediction of muscle invasiveness based on miRNA expression** [48]. The goal was to develop a model that can aid in the decision between surgical removal of the bladder to prevent invasion or bladder-conserving strategies. We first confirmed that four previously identified miRNAs effectively distinguish muscle-invasive from non-invasive bladder cancers. Based on these miRNAs, we trained highly accurate prediction models, which were evaluated on an independent patient cohort. Since the identified miRNAs are also detectable in urine samples, they may enable the prediction of bladder cancer progression in a non-invasive manner. Additionally, the small number of miRNAs might make our models more feasible for clinical applications than methods requiring hundreds or thousands of inputs, since the effort and cost of expression screening may increase with the number of measured miRNAs. Lastly, our publication strives to introduce the concept and benefits of CP to a medical readership.

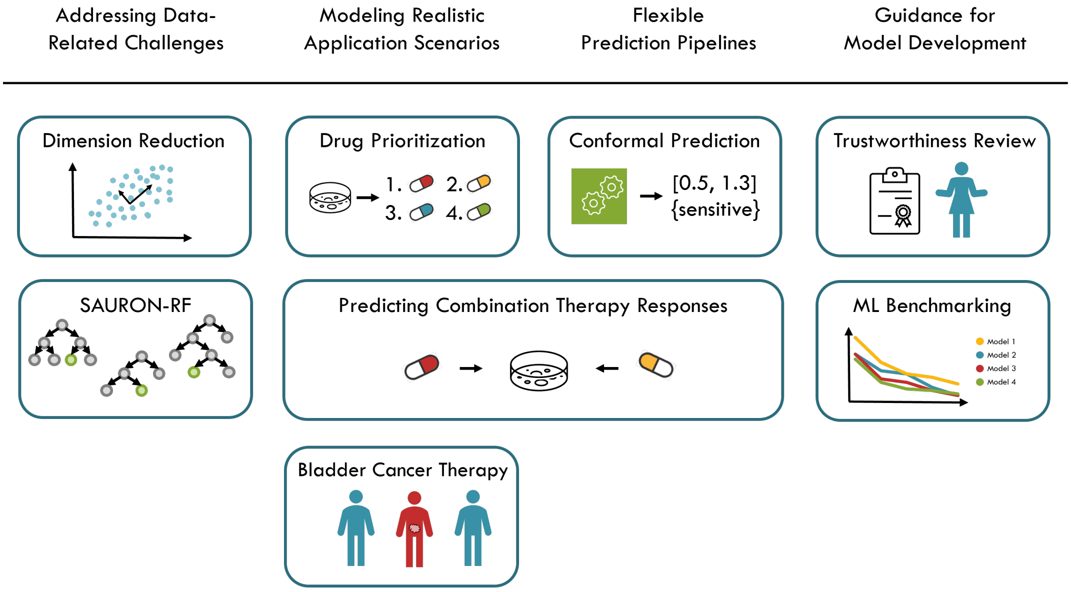


FIGURE 1.1: Thesis contributions. This figure shows the main contributions of this thesis, which are summarized in Section 1.1 and detailed in Chapters 4 to 8.

In summary, our contributions can broadly be categorized into the following four areas, each playing a crucial role in enhancing the trustworthiness of ML models and their suitability for clinical applications (cf. Figure 1.1):

1. Addressing data-related challenges: We developed highly accurate models that address challenges such as class/regression imbalance and high dimensionality, thereby outperforming state-of-the-art approaches for drug sensitivity prediction.

2. Modeling increasingly realistic application scenarios and objectives: Combining our novel sensitivity measure with our models predicting dose-specific inhibition values, we, for the first time, enable the reliable prioritization of single- and multi-drug therapies. Prioritization is a major goal of personalized medicine that goes beyond mere sensitivity prediction toward achieving actual treatment recommendations.

3. Flexible prediction pipelines: Our approaches are designed to be applicable to a broad range of prediction tasks from the realm of (drug-based) treatment optimization. For example, our CP framework is suited for any ML algorithm providing a notion of prediction uncertainty. Likewise, our models predicting inhibition values can be used to derive various (multi-)drug response measures for previously unseen cell lines and drugs.

4. Guidance for model development: Lastly, our benchmarking and trustworthiness review provide strategies on how model performance, reliability, and interpretability can be enhanced in future research.

To maximize the benefit of our work for other researchers, the code of our methods is publicly available on GitHub (https://github.com/unisb-bioinf).

1.2 Thesis Outline

The remaining chapters of this thesis are structured as follows:

Chapter 2 focuses on the development and characterization of cancer from a biological perspective. We give an overview of core cellular processes and discuss process alterations linked to cancer development. Additionally, we discuss common cancer treatment strategies and avenues on how ML can aid in combating the disease.

Chapter 3 contains a brief overview of common model systems to study drug responses, namely cancer cell lines, patient-derived xenografts, and organoids. Furthermore, the process of in vitro cell line drug screenings and the derivation of drug response measures from the screening data are described. Afterward, the *Genomics of Drug Sensitivity in Cancer* (GDSC) [49] and DrugComb [50] databases are introduced. The GDSC is one of the largest publicly available cell line databases for cancer monotherapy, while DrugComb provides a large collection of drug combination screens. We employed these databases for our analyses in Chapters 5 to 8.

Chapter 4 is concerned with the mathematical background of our research. It contains an overview of the four ML realms (supervised, unsupervised, semi-supervised, and reinforcement learning) and a detailed description of several supervised learning algorithms used throughout this thesis. Furthermore, we discuss various requirements that trustworthy ML models should fulfill, including accurate performance, reliability, and interpretability. In this context, we also describe conformal prediction in detail.

Chapter 5 presents the results of our drug sensitivity benchmarking on the GDSC database. We systematically compare four ML algorithms in combination with nine dimension reduction techniques regarding prediction performance, runtime, and interpretability.

Chapter 6 introduces our simultaneous regression and classification approach SAURON-RF and showcases its application to the GDSC database.

Chapter 7 presents our conformal prediction framework and introduces our novel drug sensitivity measure called CMax viability. We combine both with SAURON-RF to showcase how reliable drug prioritization can be performed on the GDSC database.

Chapter 8 focuses on the prediction of drug responses for combination therapy using the DrugComb database. We compare different model architectures and show how model predictions can be used to infer various measures of drug sensitivity. Additionally, we perform multi-drug prioritization.

Chapter 9 describes the application of ML and conformal prediction to predict muscle invasiveness of bladder tumors based on miRNA expression.

Chapter 10 summarizes the contributions of this thesis, discusses the shortcomings of our work, and suggests directions for future research.

Author Contributions

Many results presented in this thesis are based on joint efforts of various researchers and have already been published in peer-reviewed journals. The following chapters contain information boxes specifying author contributions and references to the corresponding publications.

Chapter 2

Cancer Biology and Treatment

Cancer has been around even before the existence of *homo sapiens*: The oldest known cancer specimen was found in a 1.7 million-year-old bone of a human ancestor [51]. The earliest written descriptions of cancer stem from Egyptian papyri dating back to 1600 BC [52]. Over a thousand years later, the term *karkinos* was first coined by Greek physician Hippocrates to describe the crab-like appearance of tumors, which was later latinized to *cancer* [53, 54]. According to Hippocrates' *theory of the four humors*, health is linked to the balance of four bodily fluids: blood, phlegm, yellow bile, and black bile [54]. It was long believed that an excess of black bile would lead to cancer [54].

Over the last centuries, our understanding of cancer has improved tremendously, often alongside technological advancements: For example, in 1863, Rudolf Virchow used the recently invented microscope to derive the cellular origin of cancer [55]. In 1775, Percival Pott discovered chimney soot as the first environmental carcinogen [56].

Following the identification of DNA as the carrier of genetic information in 1944, the breaking of the genetic code in 1961, and the sequencing of the human genome in 2000 [57], the last decades have furthered our understanding of cancer on a molecular level. We are now aware of hundreds of genes involved in cancer development, genetic predispositions, and environmental risk factors.

Historical findings have also shaped the way we treat cancer. While cancer surgery has been around for a long time, the benefits of radiation- and chemotherapy were first recognized around the year 1900, and the first targeted therapy was developed in the 1950s [58, 59]. To date, cancer research and therapy are largely supported by computers. Their continuously growing processing capabilities allow, e.g., the automated analysis of large-scale datasets and the development of advanced predictive models. Undoubtedly, these tools will continue to augment our understanding of the complex biological mechanisms behind cancer and support the development of increasingly effective treatments.

In this chapter, we summarize the current understanding of cancer biology and the current state of cancer therapy. To this end, we first describe the flow of information in healthy human cells and its regulatory mechanisms. Next, we discuss potential aberrations that alter gene expression and may lead to diseases like cancer. We then detail the process of cancer development and typical traits of cancer cells. Finally, we describe common treatment strategies and discuss how machine learning can aid in fighting the disease.

2.1 Gene Expression: The Cellular Flow of Information

Unless stated otherwise, the information described below and in Section 2.1.1 is taken from Chapters 1 to 7 of the book *Molecular Biology of the Cell* by Alberts et al. [60].

The totality of hereditary information stored in the DNA is known as the genome. Certain parts of the DNA – the genes – can be transcribed (and translated) to produce functional molecules such as proteins or functional RNAs that fulfill various roles in our cells. The process of synthesizing functional products from the information encoded in a gene is called *gene expression* and visualized in Figure 2.1. In the following, we briefly summarize the different *omics* levels that implement and regulate the flow of information from DNA to RNA and from RNA to protein: genomics, epigenomics, transcriptomics, and proteomics.

Genome level: Cells store their hereditary information as DNA (deoxyribonucleic acid) in the nucleus and (to a smaller extent) in the mitochondria. DNA is a helical molecule consisting of two paired strands of consecutive monomers called *nucleotides*. Each nucleotide consists of a sugar (deoxyribose) and phosphate backbone to which one of four bases is attached: adenine (A), cytosine (C), guanine (G), or thymine (T). In each single DNA strand, these nucleotides are covalently linked at the backbone. To form a DNA helix, two single DNA strands connect to each other via hydrogen bonds between complementary bases. This mechanism is known as *base pairing*. Base A binds with T, and C binds with G. Consequently, the two DNA strands each contain complementary sequences of nucleotides, such that each single strand can be used as a template to reconstruct the other.

The order of nucleotides in the DNA strands encodes the genetic information needed to synthesize functional products, e.g., proteins or functional RNA molecules. DNA regions that contain the blueprint for such functional products are called *genes*. Besides genes,

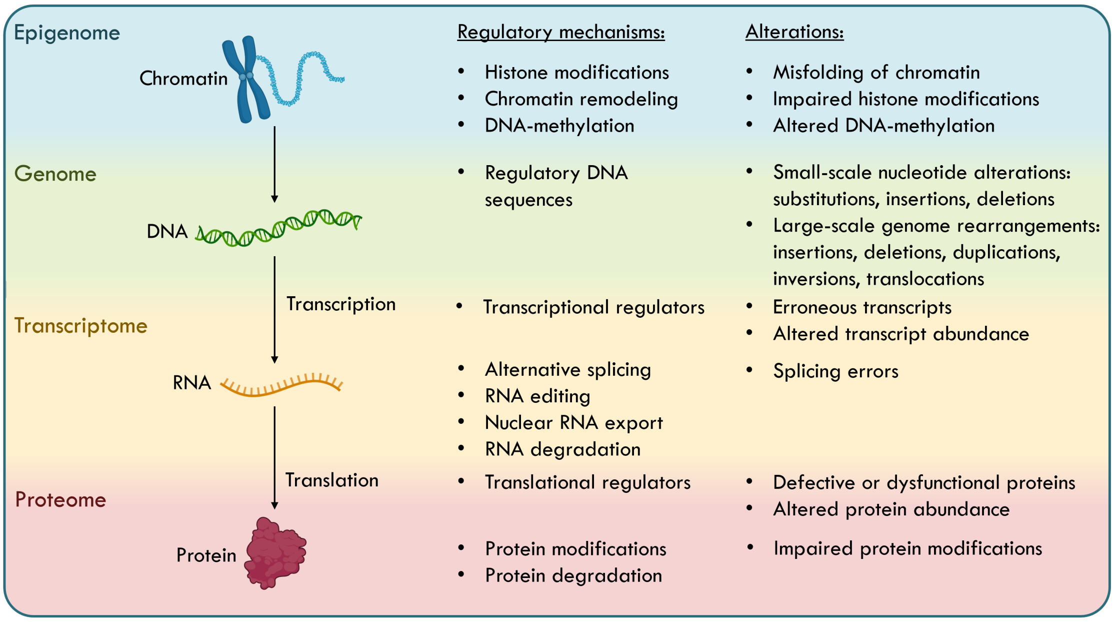


FIGURE 2.1: Flow of genetic information. This figure shows the flow of genetic information that describes how information encoded in the DNA can be expressed in the form of RNAs and proteins. Additionally, several mechanisms that regulate gene expression and alterations that may cause changes in expression and give rise to diseases such as cancer are listed. Created with BioRender.com [61].

our DNA also contains regulatory regions that control the degree of gene expression and regions whose function is currently unknown.

Epigenome level: The DNA in the nucleus is divided into 46 chromosomes and stored in a densely compacted form called *chromatin*. This packing of DNA is achieved through histone proteins around which the DNA is wrapped in a tight and ordered manner. The basic packing unit is called a *nucleosome* and consists of a histone octamer wrapped with DNA. Very densely packed chromatin is generally not accessible for DNA replication or transcription. Thus, to enable these processes, the chromatin structure must allow temporary DNA access. So-called *chromatin remodeling complexes* can reversibly modify the chromatin structure to be less (or more) tight, which is achieved, e.g., through the acetylation, methylation, and phosphorylation of histones or by repositioning nucleosomes to expose certain DNA sites [62]. Additionally, DNA methylation plays a crucial regulatory role, where methylated regions tend to be less accessible than unmethylated ones. The entirety of these modifications that regulate chromatin accessibility without altering the DNA sequence is known as the *epigenome* [63].

Transcriptome level: The first step to express the information encoded in a particular gene is transcription. Here, the gene's nucleotide sequence is read and used as a template to synthesize a complementary RNA (ribonucleic acid) molecule. The RNA is similar to the DNA but has three core differences: (1) it is a single-stranded molecule, (2) it contains the sugar ribose instead of deoxyribose, and (3) it contains the base uracil

(U) instead of thymine (T). The transcribed RNAs can serve several functions in the cell: Some RNAs function as messenger RNAs (mRNAs), which will be translated into proteins, as discussed below. However, for other genes, the RNA is the final gene product. For example, ribosomal RNAs (rRNAs) and transfer RNAs (tRNA) are crucial for translation. Other RNAs, such as microRNAs (miRNAs), regulate gene expression.

Transcription is carried out by a group of proteins called *RNA polymerases*: the RNA polymerase recognizes a so-called *promoter* sequence on the DNA that marks the starting point for transcription. To initiate transcription, the presence of many regulatory proteins, the *transcription factors*, is required. Some of these transcription factors aid in placing the RNA polymerase at the promoter and opening up the DNA double strands to begin RNA synthesis. Others serve as transcriptional activators or repressors by binding to specific regulatory sequences on the DNA. Once transcription is started, the RNA polymerase moves along the template DNA and synthesizes a complementary RNA molecule one nucleotide at a time. Eventually, the RNA polymerase encounters a terminator sequence on the DNA. At this position, transcription is ended, and the newly synthesized RNA is released.

The RNA produced by transcription is called a *transcript*, and the entirety of all transcripts present in a cell or organism (at a certain time) is called the *transcriptome* [64]. As discussed above, some RNAs (after processing) already constitute functional gene products, e.g., tRNAs, rRNAs, or miRNAs. Those RNAs that encode proteins are called *pre-mRNAs* after transcription. Before translation, they undergo several processing steps to become so-called *mature mRNAs*. This includes RNA splicing, where certain sequences are cut out of the transcript. Through *alternative splicing*, different proteins can be synthesized from the same gene by changing what parts of the transcript are removed or kept during splicing. Additionally, the RNA molecule is capped with a methylated guanine and equipped with a poly-adenine tail. These modifications prevent RNA degradation and aid its export from the nucleus to the cytosol, where the RNA can then be translated into proteins.

Proteome level: Proteins comprise long, unbranched chains of amino acids that are connected via peptide bonds. They fulfill a variety of functions in the cell: they form cellular structures, detect and relay signals from outside or within the cell, transport molecules, and catalyze a plethora of chemical reactions required for, e.g., cellular metabolism, replication, and gene expression. The term *proteome* refers to the entirety of proteins (including their modifications and interactions) present in a cell or organism (at a certain time) [65].

In the cytosol, proteins are synthesized from an mRNA template in a process called *translation*, which is carried out by ribosomal proteins: The ribosomes traverse the sequence of the mRNA three nucleotides at a time. Each nucleotide triplet (codon) encodes one of twenty amino acids. The respective amino acid is then added to a growing amino acid chain via peptide bonds. Translation starts at a specific start codon (generally sequence AUG) on the mRNA and finishes when a stop codon (UAA, UAG, or UGA) is encountered. After translation, the resulting amino acid chain folds into an energetically favorable three-dimensional conformation that gives the protein its function. Some proteins further assemble into large protein complexes, e.g., the ribosomes consist of circa eighty proteins [66]. Additionally, many proteins undergo post-translational modifications that may affect protein conformation, binding affinity, and cellular location. An example is the phosphorylation of transcription factors, which steers their transport into the nucleus [67].

2.1.1 Mechanisms of Regulating Gene Expression

Our body consists of various cell types, e.g., blood cells, skin cells, nerve cells, or cells that form specific organs. Despite their different functions and appearance, almost all cells carry identical copies of our genome and they all stem from the same origin: a fertilized egg, also known as the zygote. During embryonal development, the zygote undergoes multiple divisions, leading to the formation of embryonic stem cells. As development progresses, these cells start to differentiate into increasingly specific cell types by selectively (de-)activating certain parts of the genome and up-/downregulating the expression of specific genes. However, gene regulation is not only required in cell differentiation. It is also crucial to ensure the timely induction and correct execution of cell division. Additionally, gene regulation enables cells to react to external signals or conditions, such as the lack of nutrients or oxygen, the presence of hormones, or infections. Seven layers of gene expression regulation can be distinguished (cf. also Figure 2.1):

1. **Transcriptional control** regulates if and how often a gene is transcribed, e.g., through chromatin remodeling enzymes and through transcriptional regulators that encourage or repress transcription initiation.

2. **RNA-processing control** regulates the processing of RNA transcripts, e.g., through alternative splicing. Another control mechanism is RNA editing which alters individual positions of the nucleotide sequence.

3. **RNA transport and localization control** regulates which mRNAs are exported to the cytosol and to which locations, e.g., through mRNA export factors [68] and RNA-binding proteins that recognize specific mRNA sequences and initiate their transport to specific sites [69].

4. **mRNA degradation control** regulates which mRNAs in the cytosol should be destabilized, e.g., through shortening of the poly-adenine tail and decapping or internal RNA-cleaving through endonuclease proteins.

5. **Translational control** regulates which mRNAs in the cytosol are translated, e.g., through translational repressors that bind to the mRNA to inhibit translation or through the translation initiation complex that recruits the ribosome to an mRNA.

   Another core component of both mRNA degradation control and translational control is microRNAs (miRNAs). Since we focus on miRNAs in Chapter 9, their regulatory mechanisms will be described in more detail below.

6. **Protein degradation control** regulates which proteins should be degraded, e.g., through poly-ubiquitination that targets proteins for digestion by a protein complex called *proteasome*.

7. **Protein activity control** regulates the activation, inactivation, and cellular localization of proteins, e.g., through the binding of molecules that encourage or inhibit a protein's catalytic activity or through post-translational modifications that affect protein conformation, binding affinity, and the transport of proteins to specific cellular regions.

2.1.1.1 The Role of miRNAs in Gene Regulation

MicroRNAs (miRNAs) are a group of non-protein-coding RNAs involved in gene regulation, specifically, the targeted degradation and temporal translational inhibition of mRNAs. Currently, over 2,000 varieties of miRNAs are known [70].

To regulate gene expression, a miRNA assembles with different proteins into an *RNA-induced silencing complex* (RISC). The miRNA contained in the RISC complex can then bind to specific mRNAs through complementary base pairing using a seed sequence on the miRNA that is typically at least seven nucleotides long. Notably, a single miRNA can target different mRNAs as long as they can be recognized by the respective seed sequence. Once the RISC complex binds an mRNA, the mRNA's expression may be prevented in two ways: First, it may be cleaved by the *Argonaute* protein, which is part of the RISC complex. The cleaving exposes the mRNA to degradation through exonucleases. Alternatively, different proteins can be recruited to block translation. This often results in the mRNA being sequestered into so-called *P-bodies*, membrane-less granules that float in the cytosol. Inside the P-body, the mRNA can either be degraded or it can be stored temporarily. Since the mRNA in the P-body is isolated from the ribosomes, it cannot be translated during this storage period.

Overall, the different layers of gene regulation are intended to ensure the regular differentiation of cells and modulate their response to environmental signals and stresses. Thus, it is not surprising that defects and alterations in these processes are linked to various diseases [71–73]. In Section 2.2, we will see how the deregulation of regulatory mechanisms in cancer confers cancer cells the ability to divide uncontrollably and to escape death.

2.1.2 Genomic Alterations

Unless stated otherwise, the information presented in this section is based on Chapter 12 of the book *Human Genetics* by Ricki Lewis [74].

A multitude of genes and regulatory regions on the human DNA need to be conserved to ensure that our organism is viable and functional. Still, according to the United States National Human Genome Research Institute, the genome of a human differs, on average, at around 27 million nucleotide positions (circa 0.4% of all nucleotides) from a human reference genome [75]. While most of these variations do not (or only weakly) affect our phenotype, the remaining differences are responsible for each human's unique physical traits. Some variations mirror our ancestral history, while others are caused during our lifetime, e.g., through environmental influences. Additionally, genetic variations impact our overall health, including the susceptibility to diseases.

An alteration in our DNA sequence is called a *mutation* [76].[1] Mutations can be classified based on their acquisition, size, and effect:

Acquisition: A mutation may be either inherited or newly acquired: If a mutation takes place in the genome of a germline cell, i.e., sperm or egg cells, all somatic cells of the offspring inherit the mutation. In contrast, if a mutation occurs in a somatic cell, it will solely affect those cells that are descendants of the mutated cell, which may be only a small portion of the cells in an organism. Acquired mutations can either be caused spontaneously, e.g., through errors in DNA replication or repair, or they may be induced through external sources such as radiation or chemicals.

Size: Mutations frequently only affect a single or few nucleotides that may be substituted, deleted, or inserted at a new position. However, there also exist larger-scale genome rearrangements where entire chromosomal segments may be deleted, inserted, duplicated, or reversed.

[1] This definition is sometimes refined to state that mutations refer to alterations present in $< 1\%$ of individuals in a population to distinguish them from polymorphisms, i.e., (often harmless) genetic variants that occur in greater portions of a population.

Effect: The effects of mutations are manifold and also depend on the location of the mutation: mutations may occur in a gene or regulatory DNA sequence or regions without specific (known) functions. Large-scale mutations may also span DNA segments with various functions. Generally, any mutation may either have no effect, be beneficial, or be detrimental to the cell or organism. Small-scale mutations without apparent effect on the phenotype are called *silent*. Mutations in protein-coding genes may alter the resulting amino-acid sequence or lead to a shortened or elongated protein with potentially affected functionality [77]. Mutations in regulatory DNA regions may either enhance, inhibit, or entirely prevent the binding of interacting proteins to these regions, which can alter the expression of the regulated gene [78]. One can further distinguish between loss-of-function mutations, where the affected gene product is either absent or rendered dysfunctional, and gain-of-function mutations, where the gene product has an enhanced activity or added functionality.

Since the DNA is at the base level of the flow of cellular information, alterations in the DNA likely affect all its functional products. However, alterations can also occur at other points in the information flow as summarized in Figure 2.1: For example, errors can occur during transcription, splicing, or translation, and even correctly translated proteins may fail to fold properly [79]. Additionally, epigenetic alterations such as aberrant DNA/histone modifications can alter the chromatin state and gene expression and may increase the likelihood of genomic rearrangements [80]. Like DNA mutations, such sometimes called *epigenetic mutations* can also be inherited [81].

To avoid (genomic) alterations and their negative effects, our cells have various control mechanisms [82]: For example, several proteins can detect and repair errors in the chromatin or DNA structure, such as mismatched nucleotide pairs. Additionally, the DNA polymerases, which copy the DNA during cell replication, have various control mechanisms to ensure correct duplication. There also exist control mechanisms that govern the processing of altered gene products [79]: For example, a process called *nonsense-mediated mRNA decay* initiates the degradation of mRNAs upon detection of a premature stop codon. Additionally, aberrant proteins that fail to fold correctly can be detected and marked for degradation.

Despite these control mechanisms, mutations and other alterations may remain undetected and unrepaired in rare cases. Heritable (epi-)genomic alterations can then be passed on to the next generation of cells and may even be passed to the offspring if germline cells are affected. Unfortunately, some diseases are already caused by mutations in just a single gene. An example of such a monogenic disease is sickle cell anemia caused by a mutation in the hemoglobin gene [83]. In contrast, complex diseases such as most cancers are generally caused by a multifaceted interplay between genetic alterations and environmental factors [84, 85].

2.2 Cancer

Cancer is the umbrella term for a group of diseases characterized by the uncontrolled proliferation of aberrant cell populations. These aberrant cells developed from healthy cells over multiple rounds of cell division, accumulating heritable alterations that enable their unrestricted growth, proliferation, and survival. At their site of origin, cancer cells can form local tumors. They may also invade surrounding tissues and spread to other places in the body, interfering with the function of vital organs, blood cells, and nerves, which can eventually lead to death. For the past decades, cancerous diseases have indeed been a leading cause of death worldwide [2]. In 2022, almost 20 million new cancer cases and 10 million cancer-related deaths were reported globally [2]. In Germany alone, cancer was responsible for 21.7% of deaths in the same year [86].

In the following, we first outline the mechanisms of cancer development and discuss traits shared by most cancer types, also known as the *hallmarks of cancer*. Next, we cover several systems to classify cancer subtypes and describe the most common types of cancer therapy. Finally, we discuss how machine learning can aid in fighting the disease.

2.2.1 Cancer Development

> Unless stated otherwise, the information described in this section is taken from Chapter 20 of the book *Molecular Biology of the Cell* by Alberts et al. [60].

As discussed in Section 2.1.2, our (epi-)genome is constantly affected by various inherited, spontaneous, or induced alterations. Most of them are detected and eliminated by cellular control mechanisms. However, some alterations remain uncorrected and can be passed to the next generation of cells through cell division. Certain alterations may give a cell a selective advantage which allows it to divide more vigorously, survive longer, or live in more restrictive conditions than its neighbors. Over repeated rounds of cell divisions, alterations that are favorable for cell proliferation and survival can accumulate. This phenomenon is comparable to the natural selection in Darwinian evolution. Eventually, an entire population of aberrant cells may develop, which is then referred to as a tumor. Due to different alterations occurring in different cells, a tumor may contain multiple (epi-)genetically heterogeneous subpopulations of cells, which is one factor that complicates cancer treatment.

The different mutations found in a tumor can broadly be divided into *driver mutations* and *passenger mutations*. Driver mutations actively contribute to cancer development. In contrast, passenger mutations are simply by-products of the cancer's genomic instability. Genes that are affected by driver mutations are also known as *cancer genes* [87]. To date, over 700 cancer genes are known [88]. They can be divided into (*proto-*)*oncogenes* and *tumor suppressor genes*. Proto-oncogenes are genes that contribute to cancer development if they are mutated or their expression is increased through other mechanisms. If such a gene is mutated or overexpressed, it is then called an *oncogene*. Mutations that increase a gene product's activity or lead to the acquisition of a new functionality are called *gain-of-function* mutations. In contrast, tumor suppressor genes are genes whose inactivation contributes to cancer development. The inactivation can, e.g., occur via a *loss-of-function* mutation. Tumor suppressor genes include, e.g., genes that normally prevent unconstrained cell growth and proliferation or are involved in DNA repair [89]. In summary, cancer cells have acquired (epi-)genetic alterations whose complex interplay disables regulatory mechanisms normally at play in healthy cells. This allows the cancer to grow uncontrollably and evade cell death. It may also lead to cancer cells invading surrounding tissues and spreading to distant organs. While these are the general mechanisms of cancer development, the following section will detail specific strategies commonly exploited by cancer cells to ensure their unlimited proliferation and survival.

2.2.2 The Hallmarks of Cancer

To characterize the complex processes that guide the transformation from normal to cancerous cells, Hanahan and Weinberg proposed several *hallmarks of cancer* in the year 2000 [90]. These hallmarks define traits that cancer cells need to acquire and maintain to enable their continued survival, proliferation, and tumor spread. While the specific mechanisms and the order in which these traits are acquired can differ between tumors, it is assumed that most cancer types share these hallmarks. In 2011 and 2022, updated versions of the hallmarks of cancer have been published, currently identifying eight hallmarks that govern cancer development and progression [91, 92]:

Sustaining proliferative signaling [91]: Cells require growth signals in order to proliferate. These signals are transmitted via signaling molecules that can be detected by transmembrane receptors on a cell's surface. Healthy tissues strictly control the release of such growth signals. In contrast, cancer cells sustain their continuous proliferation in several ways: Growth factor receptors on the cell surface can be overexpressed, making cells more sensitive to growth signals. Additionally, cancer cells can produce their own growth factors, making them independent from external signaling. Furthermore, the intracellular processing of (anti-)growth signals can be altered. Lastly, cancer cells can also stimulate cells in the surrounding tissue to supply growth factors.

Evading growth suppressors [91]**:** Similar to the pro-growth signals discussed above, cell proliferation can be impeded through anti-growth molecules that bind to cell-surface receptors or through intracellular control mechanisms. Cancer cells can evade anti-growth signaling through mutation or downregulation of the corresponding receptors or tumor suppressor genes that process anti-growth signals within the cell. One control mechanism that suppresses proliferation in dense, healthy tissues, but is often absent in tumors, is *contact inhibition*: Here, neighboring cells secrete molecules that enhance cell-cell-binding and inhibit the detection of growth factors.

Resisting cell death [90]**:** Normal cells consistently monitor their intracellular conditions for any signs of abnormality, such as DNA damage, hypoxia, or hyperexpression of oncogenes. If severe issues are detected, a cell can induce apoptosis, i.e., the programmed cell death: the cell is disassembled, and its remains are consumed by neighboring cells or phagocytic cells of the immune system. Apoptosis can also be induced through extracellular signals such as the detection of specific molecules (e.g., secreted by immune cells) through cell-surface receptors. Cancer cells avoid apoptosis through diverse mechanisms that involve the inactivation or mutation of pro-apoptotic regulators and the upregulation of anti-apoptotic signaling.

Enabling replicative immortality [90]**:** After a certain number of doublings, healthy cells eventually stop dividing. A key component that controls the number of possible cell divisions is the length of telomeres, which are repetitive DNA sequences at the end of the chromosomes. With each cell division, the telomere length shortens, eventually leading to chromosome damage and cell death. In order to circumvent this replication barrier, cancer cells maintain telomere length, e.g., by upregulating the telomerase enzyme, which continuously adds nucleotides to the telomeric DNA.

Activating invasion or metastasis [90]**:** A main contributor to the deadliness of cancer is the ability of cancer cells to move away from the primary tumor site to invade surrounding tissues or travel to distant sites of the body where they may form a secondary tumor, i.e., a metastasis. To enable invasion and metastasis, mechanisms that normally tether a cell to its surroundings must be altered. This includes deregulating cell-cell adhesion proteins or integrins, which link cells to the extracellular matrix. Additionally, matrix-degrading protease proteins are commonly upregulated in cancer.

Inducing or assessing vasculature [91]**:** For access to oxygen and nutrients, tumors need to form new blood vessels through a process known as *angiogenesis*. While angiogenesis in healthy cells is only transiently activated (e.g., in embryonal development or wound healing), tumor cells permanently activate this process through deregulating angiogenesis inducer/inhibitor molecules. Additionally, cancer cells may invade surrounding tissues to access existing vasculature.

Reprogramming cellular metabolism [91]**:** To account for the increased energy demand of proliferating tumors, cancer cells alter their metabolism. In normal cells, energy is mainly generated via the *oxidative phosphorylation* pathway. Under low-oxygen conditions, cells can switch to the less effective *glycolysis* pathway. Cancer cells heavily rely on glycolysis. One reason may be the lack of oxygen often observed in tumors. However, even in oxygen-rich conditions, cancer cells may employ glycolysis since its intermediate products can be used to synthesize molecules required to sustain proliferation, such as nucleotides or amino acids.

Avoiding immune destruction [91]**:** Cancer cells use several mechanisms to avoid detection and destruction by the immune system. For example, they can secrete immunosuppressive factors or recruit inflammatory cells, both leading to a reduced or disabled immune response. Additionally, many cancer cells present specific antigens on their surface that are uniquely present or highly overexpressed in tumors [93]. The immune system uses these antigens to detect and subsequently eliminate cancer cells [93]. However, tumors can evade detection by downregulating the expression of these antigens or preventing their presentation on the cell surface [94].

In addition to these eight core hallmarks, Hanahan proposed two emerging hallmarks in 2022 that currently require further validation [92]: The first emerging hallmark called **unlocking phenotypic plasticity** proposes that cancer cells avoid terminal cell differentiation since it is typically linked to a stop of proliferation. This might be achieved through reverting, altering, or blocking differentiation processes. The second emerging hallmark called **senescent cells** covers the tumor-inhibiting and tumor-promoting properties of aging cells: Typically, senescence is linked to proliferative arrest, which should protect against tumor formation. However, there are indications that senescent cells in the tumor microenvironment might also promote tumor development. Furthermore, temporary senescence may confer treatment resistance to cancer cells.
Besides the hallmarks of cancer, Hanahan and Weinberg identified two enabling characteristics that are functionally important for cancer development and support the acquisition of the cancer hallmarks [91]:

Genome instability and mutation: Most of the traits discussed above are acquired through changes in the genome. However, there exist many mechanisms that normally ensure genomic integrity, which cancer cells need to circumvent. These involve molecules that detect DNA damage, recruit the DNA repair machinery, and repair the DNA. Additionally, cancer cells may increase their sensitivity to mutagenic agents to facilitate the occurrence of new mutations.

Tumor-promoting inflammation [91]**:** In an attempt of the immune system to attack a tumor, the tumor is often infiltrated by different types of immune cells, which causes inflammation. Paradoxically, this inflammatory state may promote tumor progression by supplying various molecules to the tumor environment: These molecules include growth factors that sustain proliferation, survival factors that limit cell death, or enzymes that modify the extracellular matrix, which can facilitate angiogenesis, invasion, and metastasis. Additionally, inflammatory cells can release reactive oxygen species, which can cause DNA mutations and, thus, contribute to genomic instability.

In 2022, Hanahan proposed two further emerging characteristics that still need to be verified [92]: The first emerging characteristic, **nonmutational epigenetic reprogramming**, proposes that not only genetic but also epigenetic alterations can further tumor development. For example, the low-oxygen conditions in tumors may lead to DNA hypermethylation. Additionally, certain histones are frequently altered in cancer. It is also assumed that various cells in the tumor microenvironment undergo epigenetic reprogramming. The second emerging characteristic, **polymorphic microbiomes**, discusses the role of tumor-promoting/-inhibiting microorganisms. For example, certain bacteria and their emitted toxins may damage DNA, cause inflammation, or stimulate cell proliferation.

2.2.3 Cancer Classification

In the previous section, we discussed several characteristics shared by most cancer types. However, it is well-known that cancer is a highly heterogeneous disease in terms of the specific molecular alterations that are causal or consequential to tumor development. Heterogeneity cannot only be observed between patients but also between different tumors of the same patient, and even between the different cells within a tumor [19]. This heterogeneity has a significant impact on treatment success [19]. It manifests, among other things, in the pace of tumor growth, the risk of metastases, or resistance toward certain drugs.

Even though each tumor is unique, several classification systems have been developed to separate tumors into distinct groups with similar properties. These classifications can aid in refining diagnoses, guiding treatment decisions, and making prognoses on disease outcome. As our understanding of cancer development and progression increases, the classifications may become increasingly fine-grained, allowing for an increasingly individualized tumor assessment and therapy.

Tissue or cell type: One relatively broad classification is based on the tissue or cell type affected by the tumor. Examples are carcinomas (originating from epithelial cells), sarcomas (cells of connective tissue), leukemias (cells of blood and bone marrow), and lymphomas (lymphatic cells). Tumors can also be classified according to their organ of origin, e.g., breast, prostate, or lung carcinoma.

Histologic tumor grade: The histologic grade measures tumor differentiation, i.e., the extent to which a tumor resembles normal tissue of the same origin [95]. Typically, one distinguishes between well-differentiated low-grade tumors that look similar to healthy tissue and poorly differentiated high-grade tumors, which typically grow faster and have a worse prognosis.

Tumor spread: The TNM classification system considers three factors: tumor size (T), the spreading to regional lymph nodes (N), and the presence of metastases (M) [96, 97]. For each factor, different stages exist, e.g., there are seven categories to classify tumor size [95]. The results of the TNM classification can then be aggregated into so-called *stage groups* that represent cases with similar prognosis [95]. These stage groups are denoted by Roman numerals from I to IV, where increasing numbers indicate a more progressed disease and, typically, a worse prognosis.

Cancer type-specific classification: Cancer type-specific classifications exist for various cancers, which may also consider genetic or molecular tumor properties. In breast cancer, for example, one can distinguish tumor subtypes based on the expression of different hormone receptors [98]. The subtypes show substantial differences in disease outcome and require distinct treatment regimens.

2.2.4 Cancer Treatment

Cancer treatment depends heavily on factors like the tumor location, size, and presence of metastases but also the general health condition of the patient. Currently, the most common types of cancer therapy are surgery, radiation therapy, and systemic therapy:

Surgery: Surgery can effectively remove localized, accessible tumors either entirely or partly. As a preventive measure, surgery can also be used to remove organs or tissues that are at risk of becoming cancerous or being invaded by a nearby tumor. However, if a tumor has already metastasized, surgery alone is insufficient. Additionally, surgeries incur risks such as wound inflammation and anesthetic complications.

Radiation therapy: Radiotherapy involves hitting a tumor with high-energy beams of radiation that damage cellular DNA, leading to cell death [99]. These energy rays are typically delivered from a radiation source outside the body. However, there also exist radiotherapies where radiating material is placed inside the body, e.g., directly at the tumor site [99]. Radiotherapy may be used for tumors that are difficult to access via surgery. It can also be applied before surgery to shrink a tumor or after surgery to eliminate potentially remaining cancer cells [10]. Even though radiation can be applied to a relatively precise area, it may still damage non-cancerous cells close to the tumor site. Common side effects of radiotherapy are fatigue, skin irritation, and nausea [13].

In contrast to surgery or radiation therapy, systemic therapy affects cancer cells at different locations in the body instead of a specific tumor site. Systemic therapy can be divided into conventional chemotherapy, targeted therapy, and immunotherapy:

Conventional chemotherapy: Since cancer is characterized by uncontrolled proliferation, conventional chemotherapeutic compounds attempt to target rapidly dividing cells by intercepting their cell cycle [100]. However, such drugs are often broadly cytotoxic, targeting not only cancer cells but any quickly dividing cells such as hair follicles or cells of the gastrointestinal tract and bone marrow [11].

Targeted therapy: Motivated by the harsh side effects of conventional chemotherapy, medical research has aimed to develop treatments that target cancer cells specifically. Such drugs directly target processes commonly altered in cancer, especially the cancer hallmarks discussed in Section 2.2.2. Specific target molecules include the protein products of oncogenes (oncoproteins) or proteins involved in the same signaling pathways as the oncoproteins.

Targeting molecules and processes that are predominantly altered in tumors reduces off-target side effects and toxicity for healthy cells. However, a drawback of targeted therapies is that they are often only effective in a small portion of cases [18]. Additionally, many targeted therapies are only transiently effective and tumors acquire treatment resistance over time [91, 101]. One explanation might be that the pathways related to a certain hallmark are partly redundant, so tumors can adapt to a treatment by activating alternative pathways instead [91]. Similarly, tumors may reconstitute the activity of disrupted pathways by altering the expression of molecules up- or downstream of the drug target [101]. Such acquired resistances can potentially be avoided by targeting multiple pathways or multiple components within one pathway simultaneously to prevent the tumor from adapting to the treatment [91, 101].

Immunotherapy: The concept behind immunotherapy is to support the body's immune system to detect and defeat cancer cells more effectively. Common examples are so-called *checkpoint inhibitors*, which block immuno-suppressive pathways that are frequently exploited by cancer cells [102] and cancer vaccines which aid the immune system in detecting tumor-associated antigens [103]. Another strategy to enhance antigen recognition is CAR T-cell therapy [104]: a patient's T-cells (which are part of the immune system) are collected and genetically altered to express a chimeric antigen receptor (CAR). The modified T-cells are then amplified and reinjected into the patient, where the added receptor aids in recognizing cancer cells.

While each of these therapy approaches has certain benefits and drawbacks, it has long been recognized that combining several treatment strategies is often beneficial. As mentioned above, surgery and radiotherapy are commonly employed subsequently. Similarly, combining radio- and chemotherapy [105], radio- and immunotherapy [106], or targeted and immunotherapy [107] are promising avenues for cancer treatment. For targeted therapies, the use of combination therapies over single-drug therapies (monotherapies) likewise has various benefits: As stated above, multi-drug treatments allow targeting multiple redundant pathways or multiple molecules in the same pathway, which can aid in overcoming acquired resistances. In general, multi-drug therapies can account for molecularly heterogeneous subpopulations within a tumor that may differ in their response to the same monotherapy. Notably, it was shown that combination treatments frequently require notably lower drug doses compared to monotherapies, which can reduce treatment toxicity (see Jin et al. [101] and Mokhtari et al. [41] for a summary of several studies). Additionally, it is still being investigated whether administering multiple drugs alternatingly may prevent tumors from becoming resistant [101]. Between 2011 and 2022 alone, the United States Food and Drug Administration approved 81 novel combination treatments for cancer, including combinations of up to four different compounds [108].

2.2.5 How Can Machine Learning Help to Fight Cancer?

Machine learning (ML) can be employed in various ways to aid in combating cancer. An extensive review on this topic was published in 2023 by Swanson et al. [109]. Below, we list the main fields of ML application in cancer and provide examples of specific use cases, some of which are addressed in this thesis.

Diagnosis: Various ML-based diagnosis tools have been proposed, which are often less invasive than conventional means [109, 110]. For example, ML-based processing of medical images can aid in localizing and monitoring a tumor and determining the tumor stage. Additionally, ML-based analysis of blood or urine samples can detect complex molecular alterations indicative of tumor presence or progression.

Prognosis: Regarding disease progression and outcome, ML could be used to assess the potential of a tumor to become invasive or form metastases [111, 112], or to predict patient survival [113]. In Chapter 9, we present classification models that should predict the potential of bladder cancers to become muscle-invasive. Based on the predicted invasive potential, the decision between surgically removing the bladder to prevent tumor spread or bladder-conserving treatments can be made.

Treatment: As discussed above, specific cancer subtypes often require designated treatment regimens. ML could aid in deriving increasingly personalized treatments by considering not only cancer types but the molecular properties of individual tumors, which are typically too complex for manual inspection. One key concept of ML-based treatment personalization is drug sensitivity prediction. It involves estimating the effect of a drug treatment on a specific tumor based on its molecular properties. Predictions for candidate drugs or drug combinations can be made to estimate whether a treatment is likely to succeed.
An advancement over mere sensitivity prediction is drug prioritization: Here, the goal is to use ML to (1) identify those drugs that will likely be effective for a given tumor and (2) to sort them by their effectiveness. This results in a personalized list of treatment options. Chapters 5 to 8 of this thesis extensively focus on the tasks of drug sensitivity prediction and prioritization. Additionally, the choice of treatment should also be influenced by potential side effects. To this end, various ML approaches for side effect prediction exist [114].

Drug Development: Various applications of ML in drug development exist, e.g., designing novel drug molecules or validating drug targets [115]. Additionally, drug sensitivity prediction can estimate the effect of experimental compounds on different types of cancer cells without the need to perform extensive screenings. This enables comparing the effects of different compounds to identify the most promising candidates that should undergo further testing or optimization.

Basic Research: Lastly, ML can enhance our understanding of the biological mechanisms underlying cancer development, progression, and treatment response. For example, one could use ML to identify cellular properties that putatively make a tumor resistant to a specific treatment or likely to have a poor outcome. Similarly, one could identify molecular properties that make a drug highly (in-)efficient for treating certain tumors. The extent to which humans can understand an ML model and its decisions is known as *model interpretability* [30–32]. Interpretable models cannot only aid basic research but may also help in explaining a models' decision-making in a clinical setting, e.g., by giving reasons for a predicted prognosis, or a treatment recommendation. We discuss general mechanisms of model interpretability in Chapter 4 and exemplarily showcase their application to cell line data in Chapters 5 and 6.

The above examples highlight the manifold ways that the fight against cancer can benefit from ML. However, many ML approaches, especially for cancer treatment, are still relatively far away from actual clinical application. The reasons will be further discussed in Chapter 10.

Chapter 3

Anti-Cancer Drug Sensitivity Testing

Studying the response of tumors to anti-cancer drug treatments is of great interest for cancer research, personalized medicine, and drug development. While the final objective is to optimally treat cancer patients and successful human trials are required for drug approval, testing newly developed drugs directly on humans has various obstacles. These include ethical concerns such as (unforeseen) treatment side effects and the high cost and time demands of human trials, which also require recruiting and retaining a sufficiently large and diverse participant pool. Consequently, earlier stages of research typically rely on model systems that mimic conditions in the human body to a sufficient degree and allow large-scale drug screenings to be performed in a time- and cost-efficient manner. Besides being easy to cultivate and screen, another desired trait for model systems is geno-/phenotypic stability, such that the effect of different drugs on the same tumor model can be investigated and connections between its geno-/phenotype and drug response can be identified.

In this chapter, we discuss how anti-cancer treatment responses can be measured using model systems for human tumors. First, we briefly discuss three prominent model systems, namely 2D cultures of cancer cell lines, patient-derived xenografts, and 3D organoid models. Since our research in this thesis is largely based on cell lines, we subsequently outline the typical workflow of a drug screening based on cell line cultures. Next, we describe how sensitivity measures for monotherapy can be derived from the screening data. We also discuss the drawbacks of commonly used measures that motivated us to develop our own sensitivity measure, the CMax viability, in Chapter 7. Afterward, we give an overview on the screening of drug combinations and introduce several synergy scores, commonly used to measure the synergistic or antagonistic interplay of drugs in a combination treatment. Finally, we introduce the two resources for cancer cell line data that we used in this thesis, namely the GDSC and DrugComb databases. The GDSC is one of the largest publicly available cell line panels for cancer monotherapy, while DrugComb is a comprehensive resource for drug combination data.

Author Contributions

The structure and content of Section 3.2 in this chapter are loosely based on my Master's thesis [116], while the text is completely rewritten. The description of synergy scores in Section 3.3.1 is based on the following manuscript (particularly, its supplement) and the corresponding preprint, which were written by me:

Eckhart, L., Lenhof, K., Herrmann, L., Rolli, L. M. & Lenhof H. P. (2025). *How to Predict Effective Drug Combinations - Moving beyond Synergy Scores.* Accepted for publication in iScience. DOI (iScience pre-proof): 10.1016/j.isci.2025.112622, DOI (bioRxiv preprint): 10.1101/2024.11.22.624812

3.1 Model Systems for Human Tumors

To study the effect of anti-cancer treatments, model systems are required that can faithfully recapitulate both the molecular properties and the drug response of tumors. Additionally, to perform machine learning (ML) using data from model systems, large datasets are required for training and evaluating prediction algorithms. Here, we discuss three standard model systems for human tumors and compare them regarding the requirements mentioned above.

3.1.1 2D Cancer Cell Line Cultures

Cancer cell lines are immortalized cancer cells that were initially derived from a human tumor and can be cultivated under laboratory conditions over prolonged time periods [117]. Conventionally, cell lines are grown in a cell monolayer on a flat surface, i.e., in 2D culture [118]. This makes the cultivation simple, and high-throughput drug screenings can be performed relatively resource-inexpensive [21]. Consequently, cell lines are a widely used model system for biomedical research, and many cell line-based drug-screening datasets are publicly available. Table 3.1 summarizes 16 of the largest cell line datasets for both mono- and combination therapies.

While several studies find cell lines to largely capture molecular properties and react to treatment according to known markers and mechanisms of drug response [119–123], others reported mixed results [124, 125]. Additionally, 2D cultures fail to capture in vivo conditions such as the three-dimensional tissue architecture, intra-tumor heterogeneity, and tumor microenvironment [126, 127]. For a comprehensive overview on (the adequacy of) using cell lines to model cancer pharmacogenomics, we recommend the reviews by Goodspeed et al. [21] and Wilding and Bogner [128].

3.1.2 Patient-Derived Xenografts

Patient-derived xenografts (PDXs) are obtained by transplanting and growing tissue from a tumor or metastasis in immunocompromised mice which can then be treated [142]. PDXs were shown to accurately recapitulate the drug response [143–145] and molecular properties of the original tumor [146, 147] while maintaining tumor heterogeneity and the tumor microenvironment to some degree [146–148]. To mimic tumor-immune interactions, the realization of *humanized PDXs* is investigated, where a human-like immune system should be generated in the mouse by engrafting haematopoietic stem cells [149]. The high effort of building and maintaining a PDX collection makes large-scale drug screening using PDXs difficult. Consequently, most studies investigate only a very limited number of PDXs or drugs, typically focused on single cancer types (see [150] for an overview). Currently, only one large-scale pan-cancer PDX screening dataset is available, namely the *Novartis Institutes for BioMedical Research PDX encyclopedia* (NIBR PDXE) [151], providing in vivo screening results for over 1,000 PDXs and 38 drugs. Note that PDXs can also be used to grow tumor cells, which are then extracted and cultivated in 2D cultures for high-throughput screens [142].

3.1.3 3D Organoid Cultures

To bridge the gap between 2D cultures and PDXs, 3D cultures of tumor tissue – also known as patient-derived organoids (PDOs) – have been developed over the last decade. In contrast to 2D cultures, where cells are grown on a flat surface, 3D cultures enable cells to assemble in three-dimensional structures, usually using a scaffold or gel matrix [149]. The 3D structure allows PDOs to capture in vivo conditions such as cell interactions [117], nutrient and oxygen gradients [117], and intra-tumor heterogeneity [152–154]. Additionally, several studies found that PDOs can model in vivo drug responses of cancer patients [155–157]. To mimic interactions with the immune system, organoids can be grown in co-culture with immune cells [149]. Despite high-throughput drug screens being simpler to perform for PDOs than PDXs, the time and cost requirements are still increased compared to 2D cultures [117]. In contrast to PDX cultures, which can be derived for a broad variety of cancer types, deriving PDOs for non-epithelial cancers (e.g., sarcomas, leukemias, lymphomas) currently remains a challenge [149]. An overview of available PDO datasets for specific cancer types can be found in [149].

TABLE 3.1: Drug screening datasets of cancer cell lines. This table provides an overview of the largest publicly available pan-cancer datasets of cell line (CL) drug screenings. For each database, the number of provided cell lines (with available screening data) and investigated compounds is denoted. For datasets providing results on combination therapy (bottom rows), the number of investigated compound combinations is listed. Additionally, we report the molecular characterizations of cell lines that are provided in each database using the following abbreviations: Exp - gene expression, Mut - mutations, CNV - copy number variations, Meth - DNA methylation, Fus - gene fusions, MSI - microsatellite instability, miR - miRNA expression, Prot - protein expression, Meta - metabolite abundance, Hist - histone modifications.

Database	# CLs	# Drugs	CL Characterization
GDSC1 [49, 129]	970	403	Exp, Mut, CNV, Meth, Fus, MSI
GDSC2 [130]	969	297	Exp, Mut, CNV, Meth, Fus, MSI
CTRPv1 [131]	242	185	-
CTRPv2 [132, 133]	860	481	Exp, Mut, CNV
PRISM primary [134]	578	4,518	-
PRISM secondary [134]	499	1,448	-
CCLE [119]	479	24	Exp, Mut, CNV, Meth, Fus, miR, Prot, Meta, Hist
gCSI [135]	410	16	Exp, Mut, CNV
Klijn et al. [136]	351	5	Exp, Mut, CNV, Fus
GSK [137]	311	19	Exp, CNV
FIMM [138]	106	308	-
NCI-60 [122]	60	$>$ 100,000	Exp, Mut, CNV, Meth, miR, Prot, Meta
DrugComb [46, 50]	2320	8,397 drugs, 739,964 combin.	-
NCI-ALMANAC [139]	60	104 drugs, 5,232 combin.	data from NCI-60
O'Neil et al. [140]	39	38 drugs, 583 combin.	-
Flobak et al. [141]	8	19,171 combin.	-

In summary, PDX and PDO model systems capture in vivo conditions such as tumor heterogeneity, tumor microenvironment, and tumor-immune interactions more accurately than conventional 2D cell cultures. Still, due to the large availability of cell line-based datasets, cell lines currently remain the most used model system for ML-based drug sensitivity prediction, even though there exist some approaches that employ PDX or PDO data [158–160]. For the analyses presented in this thesis, we focused mainly on cell line data derived from the GDSC and DrugComb datasets, which will be described in Section 3.4 and Section 3.5, respectively.

3.2 Measuring the Monotherapy Response of Cancer Cell Lines

In this section, we describe how a drug screening of 2D cancer cell line cultures is performed, including the in vitro experiments and the processing of the obtained results to generate dose-response curves. Our descriptions here are limited to monotherapy but Section 3.3 will outline how drug combinations can be screened.

3.2.1 In Vitro Workflow

The drug response of cancer cells can be assessed in vitro by treating the cells with different concentrations of a drug of interest and measuring the effect of the treatment on cell survival. Such *cell viability assays* are performed on microwell plates that are equipped with a rectangular grid of wells (i.e., small tubes) in which assays for different conditions can be performed. Typically, several cell cultures and drugs are assayed simultaneously on the same plate. In the following, we limit our descriptions to the screening of one cell line with one drug. The procedure consists of the following steps:

1. Well plate preparation [130]: A well plate is prepared with different samples, including

    - a negative control (well with untreated cancer cells in a medium, sometimes combined with a vehicle solution like DMSO (dimethyl sulfoxide) for drug dilution [130])
    - a positive control (well containing medium only)
    - several treatment samples (wells containing medium and cancer cells treated with different drug concentrations diluted in the vehicle solution)

    The number of cells in the negative control and treatment wells should be equal to assure comparability of treatment effects. To ensure a more robust screening, there are typically several positive and negative control wells, as well as several replicates for each treatment concentration.

2. Incubation: Cells are incubated for a fixed time span, usually 72 hours [49, 131].

3. Viability determination: The amount of live cells per well is quantified using different luminescent agents, depending on the assay [161]:

   - Syto60 assay: The Syto60 dye fluoresces upon binding to nucleic acids in living cells [162].
   - Resazurin assay: The redox dye resazurin can be reduced into fluorescent resorufin by living cells [163].
   - CellTiter-Glo assay: The enzyme luciferase uses ATP (adenosine triphosphate) produced by viable cells to convert luciferin into oxyluciferin under light emission [164].

   The luminescence emitted by each well is then measured, yielding intensity values that correspond to the amount of viable cells per well.

3.2.2 Processing of Intensity Values

The measured intensity values are further processed to obtain so-called *relative viabilities*. First, a background correction is performed by subtracting the intensity of the positive control (c_p) from the intensity of the negative control (c_n) and all treatment wells (t_x for all tested concentrations x) [165][1]:

$$\tilde{c}_n = c_n - c_p \tag{3.1}$$

$$\forall x : \tilde{t}_x = t_x - c_p \tag{3.2}$$

If more than one positive/negative control was present in the screening, c_n and c_p are the averaged intensities over all positive/negative controls, respectively. Similarly, replicates of the same treatment concentration are commonly averaged to obtain t_x [167, 168]. Next, relative viabilities v_x are computed by dividing the background-corrected intensities of all treatment wells by the negative control [165]:

$$\forall x : v_x = \frac{\tilde{t}_x}{\tilde{c}_n} \tag{3.3}$$

[1] Our description of the computation of relative viabilities is based on the implementation in the *gdscIC50* R package [165], which is based on a publication by Vis et al. [166]. In the publication, Equation 1 for the calculation of viabilities seems incorrect in the sense that the positive and negative controls are reversed.

Each value v_x denotes the fraction of live cells after treatment with drug concentration x compared to the untreated negative control. Typically, relative viabilities are in range $[0, 1]$, where a value of 0 indicates that all cells were killed by the treatment, whereas a value of 1 indicates that the treatment did not reduce cell growth at all. Values > 1 can occur if the treatment increases cell growth or due to experimental fluctuations. While negative viability values should generally not occur, it may occasionally happen that the background correction in Equation 3.2 becomes negative if the intensity of the positive control (c_p) is greater than that of the treatment well (t_x) in cases where the treatment killed all cells. Vis et al. clip viabilities outside the $[0, 1]$ interval before computing dose-response curves [166] (cf. Section 3.4.1).

3.2.3 Curve Fitting

The relative viabilities v_x (also known as *dose-response points*) obtained from the in vitro experiments can be used to fit a function $f(x)$ that estimates the drug response at any given treatment concentration x. Such a *dose-response curve* is typically modeled using a (four-parametric) logistic function of the following form [166, 168, 169]:

$$f(x) = \alpha_\mathrm{h} + \frac{\alpha_\mathrm{l} - \alpha_\mathrm{h}}{1 + exp(-s \cdot (x - x_\mathrm{infl}))} \tag{3.4}$$

Here, x denotes the treatment concentration, while α_h, α_l, x_infl, and s are the four curve parameters, which are visualized in Figure 3.1: $\alpha_\mathrm{h} \in (0, 1]$ and $\alpha_\mathrm{l} \in [0, 1)$ denote the curve asymptotes at high and low treatment concentrations, respectively; $x_\mathrm{infl} \in \mathbb{R}^+$ denotes the concentration at the inflection point, where the function changes curvature, and $s \in \mathbb{R}^-$ denotes the slope at x_infl.

When fitting such a logistic function to the dose-response points, the free parameters are commonly restricted to limit the flexibility of the function: Typically, α_l is set to 1 (i.e., 100% viability), indicating that at low treatment concentrations, cell viability is unaffected [166, 169]. In addition, α_h is sometimes set to 0 (i.e., 0% viability), indicating that sufficiently large concentrations will eventually kill all cells [166]. To perform the curve fitting, different approaches can be used which generally optimize the distance of the actually measured dose-response points v_x to the corresponding points on the generated curve [166, 167, 169, 170].

Two measures to assess the goodness-of-fit of a dose-response curve are the *root mean squared error* (RMSE) and the *coefficient of determination* (R^2 [171]) defined as follows: Consider n dose-response points v_{x_i} with $i \in \{1, ..., n\}$ where x_i denotes the respective treatment concentration. These points were used to fit a dose-response curve f. For each concentration x_i, the corresponding point on the fitted curve is defined as $\hat{v}_{x_i} = f(x_i)$.

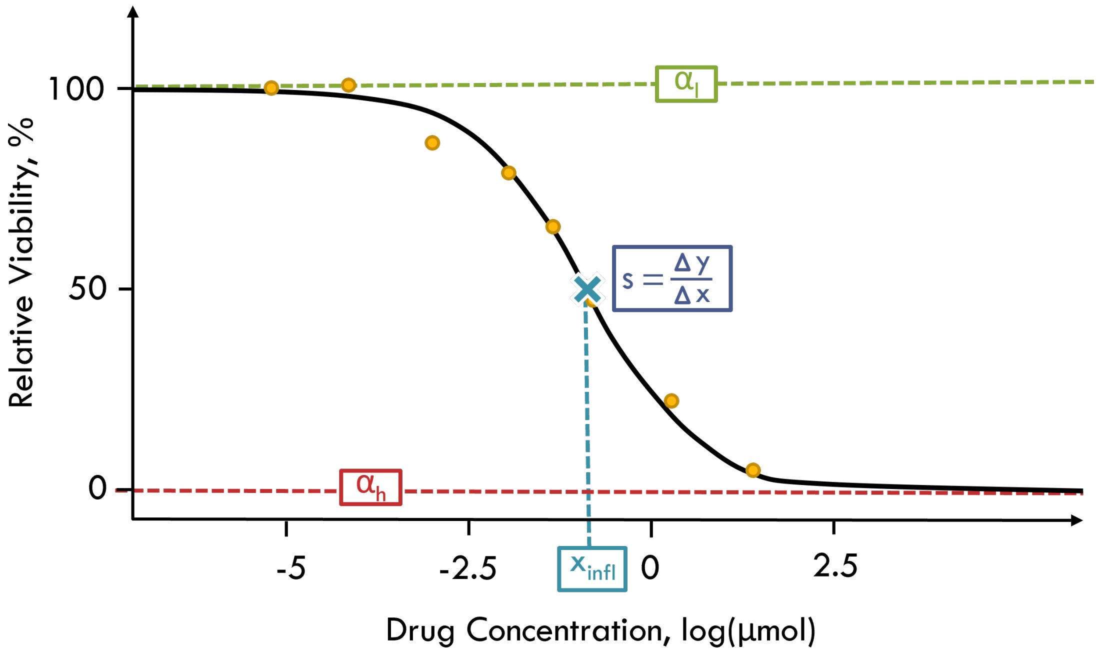


FIGURE 3.1: Dose-response curve. This figure shows a dose-response curve (black) including all curve parameters, as well as the dose-response points (yellow) that were used to fit the curve. The x-axis denotes the tested drug concentrations, and the y-axis denotes the relative viability of cells after treatment. The curve parameters include the curve asymptotes at low concentrations α_{l} (green) and high concentrations α_{h} (red), the concentration at the inflection point of the curve x_{infl} (blue) and the curve slope s at the inflection point (purple). The inflection point itself is marked by a blue cross.

The RMSE $\in [0, \infty)$ considers the distance between the actual dose-response points v_{x_i} and the corresponding points on the fitted curve $\hat{v}_{x_i}$:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{x_i} - \hat{v}_{x_i})^2} \tag{3.5}$$

The $R^2 \in (-\infty, 1]$ measure consists of two parts called the *residual sum of squares* (RSS) and the *total sum of squares* (TSS). While the RSS considers the distance between the actual points v_{x_i} and the fitted points $\hat{v}_{x_i}$, the TSS accounts for the inherent variance in the data by quantifying how much each point v_{x_i} on average differs from the mean drug response over all points $\bar{v}$:

$$RSS = \sum_{i=1}^{n}(v_{x_i} - \hat{v}_{x_i})^2 \tag{3.6}$$

$$TSS = \sum_{i=1}^{n}(v_{x_i} - \bar{v})^2 \tag{3.7}$$

$$\text{with} \quad \bar{v} = \frac{1}{n}\sum_{i=1}^{n} v_{x_i} \tag{3.8}$$

Based on these definitions, R^2 is computed as:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{3.9}$$

Vis et al. and Hafner et al. consider curves with $RMSE < 0.3$ or $R^2 > 0.5$, respectively, to be of satisfactory quality [166, 170].

3.2.4 Drug Sensitivity Metrics

Based on a fitted dose-response curve, different measures of drug response can be computed. Some of the most common measures are depicted in Figure 3.2 and discussed in the following.

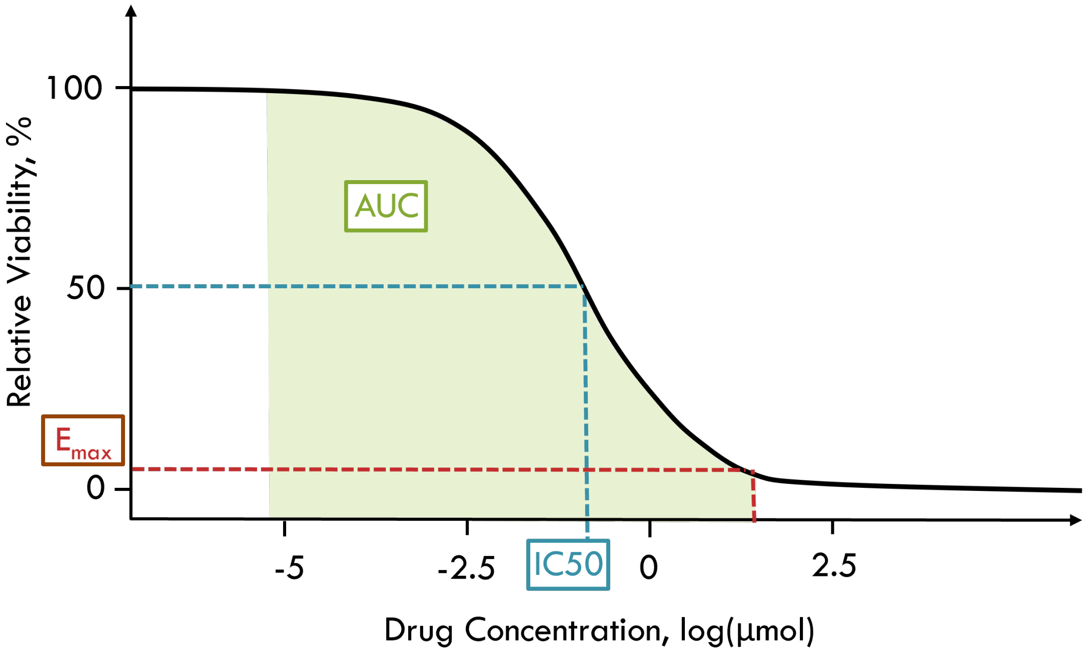


Figure 3.2: Common drug sensitivity measures. This Figure visualizes how three common measures of drug response can be derived from a dose-response curve (black). The IC50 (blue) is defined as the drug concentration (x-axis), where the curve reaches a relative viability (y-axis) of 50%. The $\mathrm{E}_{\mathrm{max}}$ (red) is defined as the relative viability at the largest screened drug concentration. The AUC (green) is defined as the area under the dose-response curve over the tested concentration range.

IC50: The IC50 value, also known as *half-maximal inhibitory concentration*, denotes the drug concentration that results in a relative viability of 50%, meaning that through treatment, the number of viable cancer cells was reduced by half [22]. If the low and high concentration asymptotes (α_{l} and α_{h}) of the dose-response curve are fixed at 1 and 0, respectively, the IC50 corresponds directly to the dose at the inflection point (x_{infl}) of the curve. However, for curves where $\alpha_{\mathrm{h}} > 0.5$, no IC50 can be computed since

50% relative viability is never reached. In such cases, the IC50 is typically set to ∞ or another large value as implemented in the code of [169] and [168].

E$_{\text{max}}$: The E$_{\text{max}}$ value denotes the relative viability reached at the highest tested drug concentration [22]. Based on E$_{\text{max}}$, another sensitivity measure called the **EC50** value can be computed as the drug concentration that results in a relative viability of E$_{\text{max}}$/2 [22].

AUC: The AUC measures the area under the dose-response curve over the tested concentration range [22]. The computed area is commonly normalized by dividing it by the area bounded by the tested concentration range on the x-axis and 1 (i.e., 100% viability) on the y-axis [166]. Based on the AUC, another sensitivity measure known as *activity area* or **AAC/AOC** (*area above/over curve*) can be computed as $AAC = 1 - AUC$ [172].

3.2.4.1 Drawbacks of Drug Sensitivity Metrics

Sensitivity metrics like IC50 and AUC are widely used to quantify drug response in cell line panels and for drug sensitivity prediction [49, 119, 132, 173]. However, they entail some drawbacks that can hamper their interpretability and their suitability for drug prioritization (cf. Section 2.2.5), which requires comparing the effectiveness of different drugs for a given sample.

Across-Drug Comparability

All of the sensitivity metrics discussed above allow for comparing the effect of the same drug on different cell lines. In this setting, a smaller IC50, E$_{\text{max}}$, or AUC always indicates a more effective treatment. However, none of these metrics is directly applicable to compare the effect of different drugs on the same cell line: The IC50 is a measure of drug concentration, but feasible treatment concentrations for different drugs can differ by several orders of magnitude [174]. Consequently, a small IC50 for one drug compared to another does not necessarily indicate a stronger sensitivity. In contrast, AUC and E$_{\text{max}}$ strongly depend on the choice of the investigated concentration range and maximum concentration, respectively: Given a dose-response curve, an arbitrary number of AUC and E$_{\text{max}}$ values can be determined by varying the considered concentration range. Consequently, AUC and E$_{\text{max}}$ are prone to be artificially increased or decreased. The extent of this in-/deflation may vary between drugs, which hampers comparisons. In

Appendix Figure B.1, we exemplarily show that the tested concentrations in the GDSC database often differ from clinically feasible concentrations, indicating that this issue is not just of theoretical nature.

To address the issue of across-drug comparability, we introduce a novel sensitivity measure called *CMax viability* in Chapter 7, which is based on clinically feasible concentrations and comparable across drugs.

Viability Bias Caused by Different Division Times

The conventional dose-response curves described above are based on the relative viability of cells as the measure of drug response. However, Hafner et al. observed that these curves are affected by the division times of the investigated cells [168, 170]. This biases metrics like the IC50 or E_{max} to be increased for cell lines with faster division times.

To correct this bias, Hafner et al. propose to generate dose-response curves based on the growth rate of cells instead of the relative viability. This allows computing response measures that are similar to the conventional IC50, E_{max} and AOC, namely the GR50, GR_{max} and GR_{AOC} [170]. However, to compute the growth rate of cells at different drug concentrations, either the cell division time or the initial cell count before treatment is required [170], both of which are often not reported [22]. In particular, they are not provided in the GDSC database, which we used for most of the analyses in this thesis.

3.2.4.2 Discretization of Drug Response Measures

All measures presented above are continuous measures of drug response. However, these measures are also commonly discretized, e.g., for the use in classification approaches for drug sensitivity prediction [36, 37, 40, 175–179] or drug prioritization [40]. The discretization typically involves computing drug-specific thresholds based on which continuous response measures can be partitioned into two classes (*sensitive* and *resistant*) or three classes (*sensitive*, *intermediate/ambiguous*, and *resistant*). For example, Knijnenburg et al. derive drug-specific thresholds from the shape of the response value distribution, where bimodal shapes indicate two distinct classes [36]. Costello et al. employ partitioning around medoids (PAM) clustering to group cell lines into three groups based on their drug response [176]. The midpoints between the cluster centroids' drug responses can then be used as thresholds. Stanfield et al. [177] consider all cell lines with IC50 below the maximum tested drug concentration as sensitive. In contrast, He et al. [178] do not employ a drug-specific, but a cell line-specific threshold, by labeling the treatments for the 50% of drugs with the smallest AUC for a given cell line as *sensitive* and the rest as *resistant*.

In the following, we describe the discretization heuristics used by Knijnenburg et al. and Costello et al. in more detail since we also utilized them in our work.

Knijnenburg et al.: We applied the approach by Knijnenburg (with some minor modifications discussed in Supplement S1 of [49]) for our method SAURON-RF (cf. Chapters 6 and 7), as well as some analyses presented in Chapter 5. For each drug, a threshold t is computed, such that cell lines with $\ln(\text{IC50}) < t$ are classified as *sensitive*, and the remaining cell lines are classified as *resistant* [36]:

1. Upsampling: For each cell line, where an IC50 value for the drug of interest is available, 1,000 data points are sampled from a normal distribution: The mean of the distribution is the ln(IC50) of the respective cell line and the standard deviation is 0.2 [49].

2. Density estimation: A kernel density estimation (KDE) with Gaussian kernel is performed to obtain a function that models the distribution of the $1000 * n$ generated data points, where n denotes the number of cell lines.

3. Modeling population of resistant cell lines: Based on the assumption that most cell lines are resistant to a drug, Knijnenburg et al. model the distribution of resistant samples as a normal distribution, where the mean corresponds to the mode (i.e., highest point) of the KDE obtained in Step 2. The standard deviation depends on whether the KDE obtained in Step 2 (1) is bimodal, (2) shows a prominent change in slope, or (3) neither of the previous two cases is true (see [36] for details). Cases (1) and (2) both aim to identify characteristics of the KDE that mark the divide between the distribution of sensitive samples and resistant samples.

4. Determination of threshold: The threshold separating sensitive from resistant cell lines is determined as the value t, where the cumulative distribution function of the distribution determined in Step 3 evaluates to 0.03 [49]. Intuitively, this marks the point where the probability of observing $\ln(\text{IC50}) \leq t$ in the distribution of resistant cell lines is 3%. Finally, we can classify all cell lines with $\ln(\text{IC50}) < t$ as *sensitive* and the remaining cell lines as *resistant*.

Costello et al.: The approach by Costello et al. relies on PAM (*partitioning around medoids*) clustering as implemented in the *cluster* R package [176, 180]. We applied this method in Chapter 7 to compute both a two-class and three-class discretization of drug responses. PAM clustering (also known as k-medoids) is an unsupervised heuristic that divides data points into k clusters (i.e., classes) [181, 182]. For our application, each data point is one cell line which is characterized by its drug response value. The aim is to minimize some distance measure between all points within a cluster and the cluster's center point, also known as the medoid. Here, the squared difference of drug response values is used as distance measure. The algorithm consists of three main steps:

1. Initialization: Greedily select k points as cluster medoids by minimizing some cost function based on the chosen distance measure. Costello et al. employ the sum of distances between each point and its closest medoid as cost [176, 180].
2. Assignment: Assign each point to the cluster defined by its closest medoid.
3. Exchange: Evaluate for each cluster if replacing its current medoid with any non-medoid point results in a reduction in cost. The exchange that results in the largest decrease in cost is then executed.

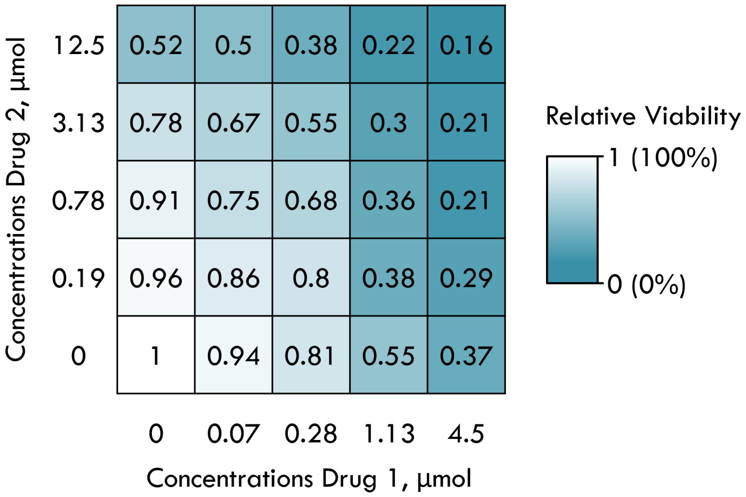
The assignment and exchange steps are repeated iteratively until no improvement in cost can be achieved anymore. The midpoints between the drug responses of the final cluster medoids can then be used as discretization thresholds.

3.3 Measuring the Drug Response for Combination Treatments

Section 3.2 was concerned with quantifying the response of cell lines to a single drug treatment, i.e., a monotherapy. However, as discussed in Section 2.2.4, combination therapy is widely used to treat cancer and is often preferred over monotherapy.

To study the effect of simultaneous treatment with two or more drugs on a cell line, the screening procedure described in Section 3.2 is slightly extended as described in the following. Here, we only consider the combination of two drugs, d_1 and d_2, but this description can analogously be extended to more drugs. Additionally, we only describe the most widely used technique for combination screening, known as a *full* or *complete* combination screen. Alternative screening procedures are discussed in [43] and [183].

In accordance with the single-drug screens presented in Section 3.2, a drug combination screening requires three types of wells: negative controls (wells with untreated cells), positive controls (wells containing medium only), and several treatment wells. Let n_1 and n_2 denote the number of non-zero treatment concentrations that should be tested for drug d_1 and d_2, respectively. The treatment wells then comprise n_1 monotherapy treatments for d_1, n_2 monotherapy treatments for d_2, as well as $n_1 \cdot n_2$ combination treatments to test the effect of combining each non-zero dose of d_1 with each non-zero dose of d_2. After the screening, relative viabilities can be computed as described in Section 3.2.2 and summarized in a *dose-response matrix*, as shown in Figure 3.3. The first row and column of the matrix correspond to the monotherapies of d_1 and d_2, respectively.



Concentrations Drug 2, µmol					
12.5	0.52	0.5	0.38	0.22	0.16
3.13	0.78	0.67	0.55	0.3	0.21
0.78	0.91	0.75	0.68	0.36	0.21
0.19	0.96	0.86	0.8	0.38	0.29
0	1	0.94	0.81	0.55	0.37
	0	0.07	0.28	1.13	4.5
	Concentrations Drug 1, µmol				

FIGURE 3.3: Example dose-response matrix. This matrix depicts the results of an exemplary drug combination screening of two drugs. Each matrix column corresponds to a constant concentration of drug 1 and each row corresponds to a constant concentration of drug 2. The matrix entries denote the relative viabilities after treatment with the respective drug concentrations. The first column and row correspond to the monotherapies with drug 1 and drug 2, respectively.

From this matrix, different measures of drug combination sensitivity can be derived. The most prominent measure is the *combination sensitivity score* (CSS) developed by Malyutina et al. [183]. This score does not consider the complete dose-response matrix but only one row and one column: For each drug, it only considers that row/column that is equal (or sufficiently close) to the monotherapy IC50 concentration of the respective drug. Based on this data, two dose-response curves are estimated: They model the combination drug response when varying the concentration of d_1 (d_2) but fixing d_2 (d_1) at its IC50 concentration. Next, the AUC values of both curves are computed (cf. Section 3.2.4) and averaged to obtain the final CSS score. Similar to the conventional AUC value for monotherapy, smaller values indicate a larger sensitivity.
However, instead of scoring combination sensitivity, it is far more common to compute so-called *synergy scores*, which will be discussed in the following.

3.3.1 Synergy Scores

Synergy scores measure the synergistic or antagonistic potential of two compounds for a given cell line by comparing their combined effect on cell viability to the expected effect obtained from a baseline model that assumes no synergism or antagonism [184]. Typically, the baseline model is derived using monotherapy data of both compounds and can be represented by a baseline dose-response matrix similar to the matrix shown in Figure 3.3. The baseline matrix and actually measured dose-response matrix are then subtracted from each other and the result is averaged over all matrix entries to obtain a final synergy score [43]. Prominent examples of synergy scores that differ solely in their computation of the baseline model are the Loewe [185], Bliss [186], HSA [187], and ZIP [184] scores that will be described in the following. For each of these scores, values > 0 indicate synergism and values < 0 indicate antagonism. Our descriptions here are largely based on a publication by Yadav et al. [184] and limited to experiments using two-drug combinations. Extensions for an arbitrary number of drugs are provided at [188].

Instead of measuring treatment effects by the *relative viability*, the literature on synergy scores typically employs the *relative inhibition* instead, which is simply defined as 1 minus the relative viability [50]. To keep the following descriptions as close to the related literature as possible, we also use the relative inhibition as measure of treatment effect here.

Consider again an experiment where a cell line c is treated with different concentrations of two drugs, d_1 and d_2. For d_1, n_1 different (non-zero) concentrations were tested, for d_2, n_2 concentrations were tested. The results of the combination treatments are reported in an $n_1 \times n_2$ matrix Y, where each entry $y_{a,b}$ denotes the relative inhibition after administering dose a of d_1 in combination with dose b of d_2. Additionally, monotherapy responses for the same n_1 concentrations of d_1 and n_2 concentrations of d_2 are required. We denote the relative inhibition obtained from the monotherapy of d_1 with concentration a as $y_{a,0}$. Analogously, the relative inhibition obtained from the monotherapy of d_2 with concentration b is given by $y_{0,b}$.

To measure the synergy between d_1 and d_2 for cell line c, the observed inhibitions $y_{a,b}$ are compared to the estimated inhibitions $\hat{y}_{a,b}$ that are calculated using a reference model which assumes no synergistic or antagonistic interaction between the two drugs. The final synergy score (SS) for a cell-drug-drug combination is then computed as the average over all concentration-specific comparisons between the observed and the estimated drug responses [189]:

$$\mathrm{SS} = \frac{1}{n_1 \cdot n_2} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} (y_{a,b} - \hat{y}_{a,b}) \tag{3.10}$$

Here, $\mathcal{A}$ and $\mathcal{B}$ denote the sets containing all tested non-zero doses for drug d_1 and d_2, respectively. For $\mathrm{SS} > 0$, the sum of the observed inhibitions is greater than the sum of the expected inhibitions, indicating synergy between d_1 and d_2. In contrast, for $\mathrm{SS} < 0$, the observed inhibitions are smaller than expected, indicating an antagonistic interaction between the drugs. In the following, we present the four most common reference models for calculating $\hat{y}_{a,b}$.

HSA [187]: The highest single agent (HSA) model expects the effect of a non-interacting drug combination to be equal to the maximum of both monotherapy effects [187]:

$$\hat{y}_{a,b}^{HSA} = max(y_{a,0}, y_{0,b}) \tag{3.11}$$

Bliss [186]: The Bliss model assumes that both drugs in a combination treatment elicit their effect independently [184]. The definition of this model is rooted in probability theory: for two independent events, E_1 and E_2, the probability of both events occurring simultaneously is given by [190]:

$$Pr(E_1 \cap E_2) = Pr(E_1) \cdot Pr(E_2) \tag{3.12}$$

Here, $Pr(E_1)$ and $Pr(E_2)$ denote the probability of each individual event occurring. Consequently, the probability of at least one of the independent events occurring is given by [190]:

$$\begin{aligned} Pr(E_1 \cup E_2) &= Pr(E_1) + Pr(E_2) - Pr(E_1 \cap E_2) && (3.13) \\ &= Pr(E_1) + Pr(E_2) - Pr(E_1) \cdot Pr(E_2) && (3.14) \end{aligned}$$

Following Equation 3.14, the Bliss score estimates the expected effect of a combination treatment as:

$$\hat{y}_{a,b}^{Bliss} = y_{a,0} + y_{0,b} - y_{a,0} \cdot y_{0,b} \tag{3.15}$$

Loewe [185]: The Loewe model, arguably, has the most complex derivation of all baseline models presented here. It is based on three core assumptions:

1. There exists a concentration A of d_1 and a concentration B of d_2, for which the monotherapy of either drug can achieve the same effect as the combination of both drugs:

$$y_{a,b} = y_{A,0} = y_{0,B} \tag{3.16}$$

2. For each concentration of drug d_1, there exists a corresponding concentration of drug d_2 that achieves the same effect. Analogously, for each concentration of d_2, a corresponding concentration of d_1 with the same effect exists.

Consequently, we can define:

$$A = a + a_b \tag{3.17}$$

$$B = b + b_a \tag{3.18}$$

Here, a_b denotes the concentration of d_1 for which $y_{a_b,0} = y_{0,b}$. Analogously, b_a denotes the concentration of d_2 for which $y_{0,b_a} = y_{a,0}$.

3. The *potency ratio* R between the two drugs is constant. The potency ratio denotes the ratio between two concentrations of d_1 and d_2 that achieve the same effect. In particular, it holds that:

$$R = \frac{A}{B} = \frac{a_b}{b} = \frac{a}{b_a} \tag{3.19}$$

Based on these assumptions, it follows that the expected combination response exhibits so-called *Loewe additivity* [184]:

$$a + a_b = A \qquad \text{(definition of } A \text{ in Equation 3.17)} \tag{3.20}$$

$$\Leftrightarrow a + b \cdot R = A \qquad \text{(solve Equation 3.19 for } a_b\text{)} \tag{3.21}$$

$$\Leftrightarrow a + b \cdot \frac{A}{B} = A \qquad \text{(definition of } R \text{ in Equation 3.19)} \tag{3.22}$$

$$\Leftrightarrow \frac{a}{A} + \frac{b}{B} = 1 \qquad \text{(division by } A\text{)} \tag{3.23}$$

Based on the Loewe additivity, $\hat{y}_{a,b}^{Loewe}$ can be computed, which requires determining the concentrations A and B, i.e., the concentrations for which the monotherapy of d_1 and d_2, respectively, achieves the same effect as the combination therapy of both drugs (cf. Equation 3.16). To estimate these concentrations, dose-response curves for both drugs need to be fit. To this end, Yadav et al. employ the following function which models the relative inhibition at concentration x:

$$g(x) = \frac{\alpha_{\mathrm{l}} + \alpha_{\mathrm{h}} \cdot \left(\frac{x}{x_{\mathrm{infl}}}\right)^s}{1 + \left(\frac{x}{x_{\mathrm{infl}}}\right)^s} \tag{3.24}$$

Note that this function differs from the dose-response curve we introduced in Equation 3.4 in two ways: First, it estimates the relative inhibition instead of the relative viability, thus, the curve is rising instead of falling as the concentration x increases. Second, a different type of function is used. Nevertheless, the same curve parameters are used in both functions: α_{l} and α_{h} denote the curve's asymptotes for low and high drug doses, respectively, x_{infl} denotes the curve's inflection point, and s denotes the slope.

Based on the curve parameters, the concentrations A and B can be estimated as [184]:

$$\hat{A} = x_{\text{infl}}^{1} \cdot \left(\frac{\hat{y}_{a,b}^{Loewe} - \alpha_{\text{l}}^{1}}{\alpha_{\text{h}}^{1} - \hat{y}_{a,b}^{Loewe}} \right)^{\frac{1}{s_1}} \tag{3.25}$$

$$\hat{B} = x_{\text{infl}}^{2} \cdot \left(\frac{\hat{y}_{a,b}^{Loewe} - \alpha_{\text{l}}^{2}}{\alpha_{\text{h}}^{2} - \hat{y}_{a,b}^{Loewe}} \right)^{\frac{1}{s_2}} \tag{3.26}$$

Here, α_{l}^{1}, α_{h}^{1}, x_{infl}^{1}, s_1 denote the estimated curve parameters for drug d_1 and α_{l}^{2}, α_{h}^{2}, x_{infl}^{2}, s_2 denote the estimated curve parameters for drug d_2. Finally, $\hat{A}$ and $\hat{B}$ can be plugged into the Loewe additivity given in Equation 3.23:

$$\frac{a}{\hat{A}} + \frac{b}{\hat{B}} = 1 \tag{3.27}$$

The estimated effect is then obtained by solving Equation 3.27 for $\hat{y}_{a,b}^{Loewe}$.

ZIP [184]: The *zero interaction potency* (ZIP) model combines ideas of both the Bliss and Loewe models. Just like the Loewe score, the ZIP score relies on accurate curve fittings for both monotherapies. It models the notion of non-interaction between drugs by assuming that the dose-response curve of one drug is unaffected by the addition of the second drug. Consequently, the combination effect can be described by simply shifting the dose-response curve of either drug by the effect of the other. Shifting the curve of drug d_1 by the effect of drug d_2 at concentration b can be described as follows [184]. For simplicity, Yadav et al. assume here that $\alpha_{\text{l}} = 0$ and $\alpha_{\text{h}} = 1$:

$$\hat{y}_{1 \leftarrow 2} = \frac{y_{0,b} + \left(\frac{x}{x_{\text{infl}}^{1}} \right)^{s_1}}{1 + \left(\frac{x}{x_{\text{infl}}^{1}} \right)^{s_1}} \tag{3.28}$$

Here, x denotes any dose of d_1. Thus, the conventional dose-response curve for d_1 (cf. Equation 3.24) is modified by simply adding $y_{0,b}$ in the numerator, thereby increasing the baseline effect. Analogously, $\hat{y}_{2 \leftarrow 1}$ can be defined as shifting the curve of d_2 by the effect of drug d_1 at concentration a. Yadav et al. show that both $\hat{y}_{1 \leftarrow 2}$ and $\hat{y}_{2 \leftarrow 1}$ are equivalent to estimating the combined drug response for concentrations a and b as follows [184]:

$$\hat{y}_{a,b}^{ZIP} = \frac{\left(\frac{a}{x_{\text{infl}}^{1}} \right)^{s_1}}{1 + \left(\frac{a}{x_{\text{infl}}^{1}} \right)^{s_1}} + \frac{\left(\frac{b}{x_{\text{infl}}^{2}} \right)^{s_2}}{1 + \left(\frac{b}{x_{\text{infl}}^{2}} \right)^{s_2}} - \frac{\left(\frac{a}{x_{\text{infl}}^{1}} \right)^{s_1}}{1 + \left(\frac{a}{x_{\text{infl}}^{1}} \right)^{s_1}} \cdot \frac{\left(\frac{b}{x_{\text{infl}}^{2}} \right)^{s_2}}{1 + \left(\frac{b}{x_{\text{infl}}^{2}} \right)^{s_2}} \tag{3.29}$$

Note that the structure of Equation 3.29 is similar to that of the Bliss score provided in Equation 3.15. Consequently, the notion of *non-interaction* between drugs in the ZIP model is also modeled through probabilistic independence [184].

Estimating the synergistic potential of compound combinations is certainly valuable for identifying promising combination treatments to undergo more detailed screening or developing novel compounds explicitly designed to work in synergy with others. However, synergy scores are difficult to employ for making personalized treatment recommendations for several reasons, which will be discussed in detail in Chapter 8. Briefly summarized, the concentration ranges and combinations that are used to compute these scores do often not correspond well to clinically feasible treatment concentrations (cf. Appendix Figure B.2) [43]. Furthermore, the underlying model assumptions (e.g., the independence assumption of the Bliss and ZIP scores or the assumption of a constant potency ratio of the Loewe score) may not hold for real-life data [44, 45]. Lastly, a high synergy between two compounds does not guarantee an overall high effectiveness of the combination treatment [46]. Thus, in Chapter 8 we suggest alternatives on how to perform drug sensitivity prediction and prioritization using data from combination screens without relying on synergy scores.

3.4 Genomics of Drug Sensitivity in Cancer (GDSC)

In this section, we introduce the Genomics of Drug Sensitivity in Cancer (GDSC) database, which we used for most of our analyses (cf. Chapters 5 to 7). The GDSC is a cooperative project between the Cancer Genome Project (Wellcome Sanger Institute, United Kingdom) and the Center for Molecular Therapeutics (Massachusetts General Hospital Cancer Center, USA). The first version of this database was published in 2012 [167], but it has been continuously expanded ever since. Currently (Release 8.5, October 2023), the GDSC includes 978 cancer cell lines that have been screened against 624 anti-cancer compounds [191], making it one of the largest publicly available pan-cancer cancer cell line panels to date (cf. Table 3.1). The screenings include clinically approved drugs, as well as drugs in clinical development and experimental compounds [49]. Cell lines in the GDSC are additionally characterized through multiple omics types, including gene expression, DNA mutations, copy number variations, DNA methylation, gene fusions, and microsatellite instability [49]. In the following, we describe the drug response and omics data provided by the GDSC in more detail. Our description of the omics data focuses mainly on gene expression since this is the main data type employed in our analyses. Details on all omics types are provided in Supplement S1 of [49]. All data can be downloaded from the GDSC website (https://www.cancerrxgene.org/downloads/bulk_download).

3.4.1 Drug Sensitivity Data

The drug response data in the GDSC can be divided into two subdatasets called GDSC1 and GDSC2. For the GDSC1, which was developed until 2015, Syto60 and Resazurin assays were used to quantify cell viability after treatment [130]. For the GDSC2, which is still under development today, CellTiter-Glo assays are used [130] (cf. Section 3.2.1). Tested drug concentration ranges were chosen by first selecting a maximum tested concentration for each drug based on in vitro experiments and clinically measured drug concentrations in human plasma [130]. This maximum concentration was then repeatedly diluted, resulting in five to nine investigated concentrations per experiment [130].
In both datasets, relative viabilities were computed as discussed in Section 3.2.2 and clipped to range $[0, 1]$. However, fluorescence intensities of replicate experiments using the same drug concentration were not averaged, but relative viabilities were computed for each replicate separately. Consequently, if replicate analyses were performed, several dose-response points for the same drug concentration are reported and used for the curve fitting [130].
To fit dose-response curves, a multilevel mixed effects model by Vis et al. [166] was applied to the GDSC1 and GDSC2 data separately [130]. The curve asymptotes (cf. Equation 3.4) were fixed at $\alpha_{\mathrm{l}} = 1$ and $\alpha_{\mathrm{h}} = 0$. Additionally, the approach by Vis et al. assumes that the curve slope s is constant across all experiments involving the same cell line. Fitted curves where the RMSE between the actual dose-response points and the corresponding points on the fitted curve was greater than 0.3 were deemed poor quality and discarded [192].
From the fitted curves, IC50 and AUC values were derived as discussed in Section 3.2.4. The IC50 values are provided in a logarithmized form (natural logarithm), while AUC values were normalized to the range $[0, 1]$ by division with the area bounded by the tested concentration range on the x-axis and 1 on the y-axis [192].

3.4.2 Gene Expression Data

Gene expression was measured using the *Affymetrix Human Genome U219* microarray. Microarrays detect expression through hybridization of fluorescence-labeled cDNA probes (e.g., derived from the RNA extracted from a cancer cell line) to complementary DNA sequences attached to a well plate. The position of each spot on the plate allows the identification of the corresponding transcript. After washing away unbound probes, the amount of fluorescence emitted at each spot is proportional to the amount of bound probe, i.e., the probe's expression.

The raw fluorescence signals were processed using the *robust multi-array analysis* (RMA) algorithm as described by Irizarry et al. [193], resulting in expression values for 18,562 loci (cf. Supplement S1 of [49]).

3.4.3 Further Omics Data Types

The GDSC additionally provides the following types of omics data (see Supplement S1 of [49] for a detailed description of the corresponding experimental procedures):

- Whole exome sequencing (*Agilent SureSelectXT Human All Exon 50Mb bait set*)
- Copy number alterations (*Affymetrix SNP6.0 Array*)
- DNA methylation (*Illumina Human Methylation 450 Array*)
- Gene fusions (targeted *polymerase chain reaction* (PCR) sequencing or split probe *fluorescence in situ hybridization* (FISH) analysis)
- Microsatellite instability (based on presence of insertions or deletions in five marker microsatellites)

3.5 DrugComb

While the GDSC is a relatively large resource for monotherapy data, it does not provide any data on drug combination screens. The currently largest resource for combination data is the DrugComb data portal, which was introduced in 2019 [46, 50]. DrugComb contains harmonized results of drug screens from different sources. To date, around 750,000 drug combinations derived from 37 datasets are available in DrugComb [46], including the National Cancer Institute ALMANAC [139], AstraZeneca DREAM challenge [194], and O'Neil [140] drug combination datasets (cf. Table 3.1). DrugComb also encompasses monotherapy data, including data derived from the GDSC. An overview on all datasets can be found in the supplement of [46].

The data from various sources is harmonized by (1) standardizing cell line and drug names, (2) using a common file format for all data sources, and (3) providing all screening results as *relative inhibition* values. The relative inhibition of a treatment is defined as 1 minus the relative viability (cf. Equation 3.3). In addition, measures of drug sensitivity and drug synergy are provided, including the CSS score [183], as well as the Loewe

[185], Bliss [186], HSA [187], and ZIP [184] synergy scores.

While DrugComb does not provide any omics-characterizations of the screened cell lines, they list each cell line's COSMIC ID. These unique identifiers are also used in the GDSC database, so omics-measurements can be obtained from there. All data are available on the DrugComb website (https://drugcomb.org/), which also encompasses different means for data visualization and analysis, including ML models for the prediction of sensitivity and synergy scores.

Chapter 4

Machine Learning

Nowadays, artificial intelligence (AI) is omnipresent in our everyday lives: when we look at our smartphones, we are shown personalized advertisements or social media posts and AI-based weather forecasts. Many homes are equipped with smart devices that automatically regulate the room temperature, change lighting, or reorder groceries. Companies use AI to optimize manufacturing, monitor inventories, or automate customer support. Often, we may not even notice how much AI is involved in our daily routine. However, the recent advancements in generative AI, such as the large language model ChatGPT [195] or the image generator DALL·E [196], allow even laypeople to directly interact with AI models to produce text, images, or music.

The examples above give a glimpse into the remarkable progress AI has made since its birth in the 1950s [197]. However, current AI models are far from infallible: ChatGPT, e.g., frequently provides wrong source citations [198], while image generators are notoriously unable to accurately represent human anatomy like hands or teeth [199]. Even though a missing finger on an AI-generated image might be just a slight inconvenience, other applications for AI, such as autonomous driving, cannot allow any mistakes without risking serious consequences.

In this thesis, we apply AI – specifically machine learning (ML) – to biomedical data with the final goal of building models that should eventually be used in healthcare to optimize cancer treatment. Undoubtedly, the healthcare sector can benefit immensely from AI, e.g., through image-based tumor detection, surgical robots, or AI-based systems for medical decision support [200]. However, healthcare is another field where mistakes are hardly tolerable. For example, in 2019, a study found that a commercial algorithm for health risk assessment systematically underestimated the need for care of black compared to white patients [201]. Consequently, black patients were less frequently included in high-risk care programs. With around 200 million people being screened by comparable tools in the USA every year [201], this exemplifies how inadequate AI systems can cause significant harm if not designed, trained, tested, and deployed appropriately, i.e., if the systems are not trustworthy.

In this chapter, we give an overview of the fundamentals of ML and discuss requirements that ML models should fulfill to be trustworthy for application in sensitive fields such as healthcare. We first define the term *machine learning* and its four major realms, namely supervised, unsupervised, semi-supervised, and reinforcement learning. Next, we describe several supervised learning algorithms that were employed for the analyses presented in Chapters 5 to 9. Finally, we detail requirements for trustworthy AI, including performance, reliability of individual predictions, and model interpretability. In this context, we also discuss how to tailor models to a certain task (here: drug sensitivity prediction) and highlight factors that may negatively impact trustworthiness.

Author Contributions

The structure and content of Sections 4.1 and 4.3 in this chapter are loosely based on the publication below. In particular, the taxonomy and literature research on model interpretability presented in Section 4.3.3 were first published therein.

Lenhof, K., Eckhart, L.*, Rolli, L. M.*, & Lenhof, H. P. (2024). *Trust me if you can: a survey on reliability and interpretability of machine learning approaches for drug sensitivity prediction in cancer.* Briefings in Bioinformatics, 25(5), bbae379. DOI: 10.1093/bib/bbae379 (* authors contributed equally)

The majority of the research presented in this publication was conducted by Kerstin Lenhof who also drafted the manuscript. Lisa-Marie Rolli and I were involved in conducting the extensive literature research presented in the publication and edited and reviewed the manuscript.

The description of neural networks in Section 4.2.5 is loosely based on my Master's thesis [116]. The text is, however, completely rewritten.

The description of dimension reduction in Section 4.3.4.1 is based on the following publication, which was written by me and is discussed in detail in Chapter 5:

Eckhart, L., Lenhof, K., Rolli, L. M., & Lenhof, H. P. (2024). *A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.* Briefings in Bioinformatics, 25(4), bbae242. DOI: 10.1093/bib/bbae242

4.1 The Four Realms of Machine Learning

While AI is a wide field, this thesis is only concerned with one subdomain of AI, namely *machine learning* (ML). El Naqa and Murphy collected several definitions for the term *machine learning* [202], which can be summarized as developing algorithms that learn to optimize their performance for a certain task from *experience*, e.g., in the form of example data. While most AI systems arguably fall into the category of ML, there also exist methods that, instead of learning from data, are explicitly programmed to achieve a certain task, e.g., through following human-defined rules.

ML can broadly be divided into four main realms, each suited for varying applications: *supervised*, *unsupervised*, *semi-supervised*, and *reinforcement learning*. In the following sections, we will introduce each realm and briefly discuss its relevance for drug sensitivity prediction. Since this thesis focuses primarily on supervised learning, this realm will be explained in most detail.

4.1.1 Supervised Learning

To understand the concept of supervised learning, we first introduce several definitions and notations: Consider an $N \times P$ matrix $\mathbf{X}$ containing observations of P feature variables $X_1, ..., X_P$ for N data points (samples). Each matrix row $x_i \in \mathcal{X}$ corresponds to a vector containing the observed values of sample i for all P features. Additionally, we consider an N-dimensional response vector $\mathbf{y}$, where each entry $y_i \in \mathcal{Y}$ denotes the value of response variable Y for sample i. The same data can alternatively be described by a set of feature-response pairs $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where each pair corresponds to one sample [203]. It is typically assumed that these pairs are drawn i.i.d. (independently and identically distributed) from a common distribution in the range $\mathcal{X} \times \mathcal{Y}$ [203]. The term *identical* refers to all pairs being drawn from the same probability distribution, while the term *independent* requires that the feature values x_i of one data point do not influence the feature values of another.

The goal of supervised learning is to derive a model that can predict the value of the response based on the values of the features (also known as *predictors*). To this end, we assume that there exists a mapping between the features $X = (X_1, ..., X_P)$ and the response Y that can be described using an unknown but fixed function $f : \mathcal{X} \to \mathcal{Y}$ [204]:

$$Y = f(X) + \epsilon \tag{4.1}$$

Here, ϵ is a random error term, which is assumed to be independent of X and to have mean 0. A supervised ML model can be seen as a function $\hat{f}$ that approximates f based on the observed data pairs. Using $\hat{f}$, predictions can then be made for data points where

no response may be available. If the response is continuous, estimating f is referred to as a *regression* task; for a discrete response, it is called *classification*.
Most drug sensitivity prediction approaches fall in the category of supervised learning (see Table 4.1 in Section 4.3.3). Here, the features are typically multi-omics measurements of cancer cell lines, and the response is some measure of drug response, e.g., the IC50 value (cf. Sections 3.2.4 and 3.2.4.2).
When estimating f, one is typically driven by two motivations called *prediction* or *inference* [204]. In the prediction setting, the aim is to estimate f as accurately as possible in order to make precise predictions for novel data points with unknown responses. For a novel data point x, its response can be predicted as:

$$\hat{y} = \hat{f}(x) \tag{4.2}$$

In the inference setting, the aim is to better understand the relationship between the predictors and response by investigating the shape of $\hat{f}$. This includes, e.g., the identification of predictors that have the strongest impact on the predicted response or studying the relationship between individual predictors and the response. Often, more complex models are able to produce more accurate predictions, making them well-suited for the prediction task. However, their complexity makes it difficult for humans to interpret the learned relationship between features and response. In contrast, simpler (e.g., linear) models are often better suited for inference but may yield less accurate predictions. In practice, however, one is often interested in a model that is suited for both prediction and inference: For drug sensitivity prediction, for example, a common aim is not only to accurately estimate the effect of a drug treatment on a cell line but also to infer features (biomarkers) that are frequently associated with treatment resistance or sensitivity. Similarly, for the application of drug sensitivity prediction models in clinical decision support tools, the models must be highly accurate but should also allow to derive explanations for a certain prediction. In Section 4.3, we further elaborate on model requirements from the viewpoint of trustworthiness.

So how can f be estimated? In Section 4.2, we will discuss several ML algorithms for this purpose in detail. They all have in common that they require two disjoint datasets containing samples with known features and known response called the *training data* and the *test data*. While the details are described below, the purpose of these datasets can be briefly summarized as follows: The training data is used to optimize $\hat{f}$ (i.e., to *train* the model) by minimizing the deviation between the actual and the predicted responses of the training samples. After $\hat{f}$ is determined, the test data is used to quantify how accurate the model's predictions are for samples that were not involved in the training process. The training and test data are obtained by separating all available samples (i.e., the samples contained in the predictor matrix $\mathbf{X}$ and response vector $\mathbf{y}$) into two disjoint datasets.

When fitting an ML model, the difference between actual and predicted responses of the training samples is typically measured using a so-called *loss function*. For regression, a common loss function is the mean squared error (MSE). As indicated by its name, the MSE measures the average squared difference between the actual and predicted responses. For a training set consisting of N_{train} samples, the MSE is given by:

$$\text{MSE} = \frac{1}{N_{\text{train}}} \cdot \sum_{i=1}^{N_{\text{train}}} (y_i - \hat{y}_i)^2 \tag{4.3}$$

When the aim is to minimize the MSE, the best predictions are achieved for [205]

$$\hat{f}(x) = E[Y|X = x] \quad , \tag{4.4}$$

where E denotes the conditional expected value that should be estimated using the training data. Based on this observation, two strategies for supervised learning can be distinguished: (1) *discriminative* models aim to estimate the conditional density $p(y|x)$ directly, whereas (2) *generative* models estimate the joint density $p(x, y) = p(x|y) \cdot p(y)$, which can then be used to derive $p(y|x)$ using Bayes' Theorem [203].

Even though $\hat{f}$ is optimized using the training data, we ultimately desire a model that is able to make accurate predictions for novel, unseen data points (i.e., data points that were not used to estimate $\hat{f}$) such as the samples in the test dataset. To quantify the average *test* or *generalization error* [206] of a model, the model is used to make predictions for each sample in the test data and a loss function such as the MSE is evaluated to compute the overall test error.

To gain a better understanding of the factors that impact prediction quality for unseen samples, consider the following equation, which describes the expected squared error for an unseen test sample x:

$$E[(y - \hat{f}(x))^2] = Var(\hat{f}(x)) + Bias(\hat{f}(x))^2 + Var(\epsilon) \tag{4.5}$$

Equation 4.5 is also known as the *bias-variance-decomposition* [207]. It reveals that the expected squared error can be decomposed into three components [207]:

1. $Var(\hat{f}(x))$ denotes the variance of the model. It describes the amount by which $\hat{f}$ may change if our model was trained using a different training dataset. Ideally, the final model should remain relatively consistent for varying training data, so in the best case, this quantity should be 0.

2. $Bias(\hat{f}(x))^2$ refers to the bias that is introduced by the choice of model class we

employ to derive $\hat{f}$: For example, the class of linear models (cf. Section 4.2.1) restricts $\hat{f}$ to be linear, assuming a linear relationship between features and response. Ideally, the bias should be 0. If however, the chosen model class is not suited to model the problem at hand (e.g., because the true relationship between features and response is non-linear), the bias increases.

3. $Var(\epsilon)$ denotes the variance of the error ϵ. This includes factors such as noise in our data but also accounts for unknown factors that might impact the response but are not measured by our predictors [204].

The terms $Var(\hat{f}(x))$ and $Bias(\hat{f}(x))^2$ are also referred to as the *reducible error* since $\hat{f}$ can be optimized to minimize these quantities through selection of an appropriate model class and a sufficiently large number of training samples. In contrast, $Var(\epsilon)$ is referred to as the *irreducible error* since this error cannot be reduced by changing the model class or gathering more samples. This means that the prediction error can never be below $Var(\epsilon)$ (unless other/further features are considered). Consequently, to minimize the error, we can minimize $Var(\hat{f}(x))$ and $Bias(\hat{f}(x))^2$.
Unfortunately, models with small variance generally suffer from an increased bias and vice versa, which is also known as the *bias-variance trade-off* [207]: Models that make few assumptions about the relationship between features and response tend to have a low bias because the estimated function $\hat{f}$ can be highly flexible. For example, there exist neural network architectures that can theoretically model any continuous, real-valued function, even very complex ones [208, 209]. Allowing $\hat{f}$ to be highly complex can, however, increase variance and lead to overfitting, where the model makes very accurate predictions for the training data but fails to generalize well to previously unseen samples. Thus, overfitting is typically characterized by a small training error but a large test error. In contrast, models that heavily restrict $\hat{f}$ (e.g., by assuming the function to be linear) are less prone to overfitting and thus show a lower variance. However, this restriction of $\hat{f}$ can lead to an increased bias, which typically manifests in an underfitting of the data. In this case, the model is too rigid and often too simple to model the training data sufficiently well, and thus, it may also fail to generalize well to unseen data.

Above, we described how the performance of a trained model for unseen cell lines can be assessed using a dedicated test set of samples that were not used to train the model. Another method to estimate the generalization error of a model that is statistically more sound than just a single training and test set is *k-fold cross-validation* (CV): Here, the available data is split into k disjoint subsets (i.e., *folds*) of approximately equal size. Next, $k-1$ folds are used for model training, while the remaining fold is used for testing. This process is repeated k times, such that each fold was used for testing once. The final CV error is then computed as the average test error over all k test folds.

4.1.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning also comprises a feature matrix $\mathbf{X}$ but lacks any corresponding response vector $\mathbf{y}$. Consequently, unsupervised learning is not concerned with making predictions but rather with deriving knowledge from the structure of the data, i.e., inferring properties of the underlying density $p(x)$ [210]. Common methods of unsupervised learning include principal component analysis (PCA), clustering algorithms, and association rule mining [210].

While the prediction of drug responses directly cannot be modeled as an unsupervised task, unsupervised methods can still be useful to study drug responses. For example, PCA can be applied to reduce the dimension of model inputs while still capturing the most relevant information in the data (cf. Chapter 5). Clustering can be applied to divide samples into groups to identify common feature patterns and distinguishing characteristics. In Chapter 7, we use clustering to derive sample labels for the classification of drug responses.

4.1.3 Semi-Supervised Learning

Semi-supervised learning combines properties of both supervised and unsupervised learning. It describes a scenario where the available feature matrix $\mathbf{X}$ can be divided into two submatrices: $\mathbf{X_r}$, where the corresponding response values are given by $\mathbf{y_r}$, and $\mathbf{X_u}$, where no responses are available [211]. While data points in $\mathbf{X_u}$ could simply be discarded to obtain a supervised learning scenario, they might still provide valuable information for model development: Semi-supervised learning is based on the assumption that the feature distribution $p(x)$ contains information that may help estimate $p(y|x)$. Thus, a better characterization of $p(x)$ using the unlabeled data should yield a better model [212]. Based on this idea, further assumptions have been derived for semi-supervised learning (mainly classification) [212]:

- Smoothness assumption: If two data points are close to each other in the feature space, their corresponding outputs should also be close.
- Low-density assumption: A classifier's decision boundary (i.e., the area in the feature space where the prediction changes from one class to another) should not pass through regions of high data density.
- Manifold assumption: The high-dimensional data space is composed of lower-dimensional subspaces (manifolds) in which all data points lie. Furthermore, data points in the same manifold should share the same class.

Consequently, making use of the unlabeled samples can be beneficial to better characterize regions of high/low data density and, in turn, to estimate a suitable decision boundary. One commonly distinguishes between *transductive* and *inductive* semi-supervised methods [211]: Transductive methods aim to generate labels for the unlabeled data points. Subsequently, the now entirely labeled data may be used to train a supervised ML model. In contrast, inductive methods aim to develop a prediction model that learns from both the labeled and unlabeled data points without generating labels for the latter. For drug sensitivity prediction, we are only aware of one inductive semi-supervised approach by Rampášek et al., where a gene expression embedding is constructed using cell lines with and without available drug responses [213].

4.1.4 Reinforcement Learning

In contrast to the previously presented realms, reinforcement learning does not rely on a predefined set of data points for learning but instead learns from interaction: It is typically modeled through an agent that performs certain actions in a dynamic environment to achieve some task [214]. Each performed action changes the state of the environment, and these changes are reported back to the agent. Additionally, a reward or penalty is provided that scores the current action with respect to its benefit for achieving the desired task [214]. An example would be a robot (agent) located in a factory hall (environment) where it should learn to transport goods from one position to another (task) by moving in different directions (actions). The agent should learn to perform actions that maximize the long-term rewards, e.g., a successful and quick delivery of goods. Since multiple subsequent actions are typically required to fulfill a task and the immediate reward of an action might not correspond directly to its long-term benefit, the agent is required to explore the effect of its actions through trial and error [214].

Even though the reported rewards can be seen as a form of supervision, the dynamic generation of new information to learn from distinguishes reinforcement learning from supervised learning, where training is based on a fixed set of predetermined data points. For drug sensitivity prediction, we are only aware of one method that could be interpreted as reinforcement learning: In PPORank by Liu et al. [215], an agent performs drug prioritization by assembling a ranking of drugs from most to least effective for a given cell line. The ranking is constructed by choosing one drug at a time starting with the most effective. For each choice, the agent achieves a reward based on how effective the chosen drug is compared to the optimal choice.

4.2 Supervised Learning Algorithms

In this section, we describe several popular groups of ML algorithms in detail, which we employed for the analyses presented in this thesis. For simplicity, we assume all input features to be continuous since the models presented in this thesis only use continuous inputs. Consequently, this chapter will use the following notations: The entire training dataset is denoted as $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$. Here, $\mathbf{X} \in \mathbb{R}^{N \times P}$ is the input feature matrix, where each column corresponds to an input feature X_j with $j \in \{1, ..., P\}$. Each row vector $x_i = (x_{i1}, ..., x_{iP})$ corresponds to the feature values of sample $i \in \{1, ..., N\}$. Furthermore, $\mathbf{y} = (y_1, ..., y_N)^T$ denotes the N-dimensional vector that encodes the response variable Y. The entries of $\mathbf{y}$ are continuous values for regression ($\mathbf{y} \in \mathbb{R}^N$) and discrete values for classification ($\mathbf{y} \in C^N$, where $C = \{c_1, ..., c_K\}$ denotes a set of K classes).
Additionally, $\mathbf{w} = (w_1, ..., w_N)^T \in \mathbb{R}^{+N}$ is an N-dimensional vector of non-negative sample weights. These weights can be used to increase the importance of certain samples in the training process by assigning them a larger weight. While sample weights are generally not required to train an ML model, they play a crucial role in our SAURON-RF model discussed in Chapters 6 and 7, where they are used to improve predictions for drug-sensitive samples. Hence, we include them in our descriptions here. Setting all sample weights to the same value (e.g., $1/N$) results in a model where all samples are equally important.

4.2.1 Linear Models

Linear models assume that the relationship between the predictor variables $X_1, ..., X_P$ and the response variable Y is linear, i.e., the response can be modeled as a linear combination of the features (and an error term ϵ) [216]:

$$Y = \beta_0 + \left(\sum_{j=1}^{P} \beta_j \cdot X_j \right) + \epsilon \tag{4.6}$$

Here $\beta_0, \beta_1, ..., \beta_P$ are feature coefficients that need to be approximated such that a prediction for a sample x_i can be computed as:

$$\hat{f}(x_i) = \hat{\beta}_0 + \left(\sum_{j=1}^{P} \hat{\beta}_j \cdot x_{ij} \right) \tag{4.7}$$

The estimated coefficients $\hat{\beta}_1, ..., \hat{\beta}_P$ can be derived by minimizing the squared difference between the actual response and the predicted response over all N training samples [216]:

$$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_P)^T = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} w_i \cdot (y_i - \hat{f}(x_i))^2 \right\} \tag{4.8}$$

$$= \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} w_i \cdot \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j \cdot x_{ij} \right)^2 \right\} \tag{4.9}$$

Here, w_i denotes a sample-specific weight, which is typically set to 1 such that all samples are weighted equally. An issue with this model is that the estimated coefficients can suffer from high variance when highly correlated, i.e., interchangeable, features exist which has a negative impact on predictions (cf. bias-variance trade-off in Equation 4.5). To counteract this problem, so-called *shrinkage* or *regularization methods* have been developed. These approaches constrain the size of coefficients by shrinking them towards zero, leading to a reduction in variance at the cost of a (typically smaller) increase in bias [216].

One regularization method called *ridge regression* was proposed by Hoerl and Kennard [217]. For ridge regression, the coefficients are chosen to minimize [216]

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \left(\sum_{i=1}^{N} w_i \cdot \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j \cdot x_{ij} \right)^2 \right) + \lambda \cdot \sum_{j=1}^{P} \beta_j^2 \right\} \quad , \tag{4.10}$$

where $\lambda \geq 0$ is a tuning parameter that controls the impact of the regularization penalty. As penalty, the squared L2 norm is used, which causes coefficients to be shrunken towards 0. However, using this norm, coefficients can never become exactly 0, such that the corresponding features are excluded from the model. Therefore, an alternative regularization method based on the L1 norm was developed by Tibshirani [218]. This method is called the *least absolute shrinkage and selection operator*, or short *lasso*, and determines coefficients by minimizing [216]:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \left(\sum_{i=1}^{N} w_i \cdot \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j \cdot x_{ij} \right)^2 \right) + \lambda \cdot \sum_{j=1}^{P} |\beta_j| \right\} \tag{4.11}$$

Using the L1 instead of the L2 norm enables coefficients to become exactly 0. Thereby, the corresponding features are excluded from the model. However, the lasso also has some drawbacks [219]: (1) If highly correlated variables exist in the data, the lasso tends to select only one of them arbitrarily while excluding the others from the model. (2) For $P > N$, the lasso model can select at most N features with coefficients $\neq 0$. Thus, potentially relevant features might not be included in the model.

To exploit the respective benefits of ridge regression and lasso while simultaneously evading their respective drawbacks, Zou and Hastie propose a combination of both methods, which is known as the *elastic net* [219, 220]:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \left(\sum_{i=1}^{N} w_i \cdot \left(y_i - \beta_0 - \sum_{j=1}^{P} \beta_j \cdot x_{ij} \right)^2 \right) \right. \\ \left. + \lambda \cdot \sum_{j=1}^{P} (\alpha \cdot \beta_j^2 + (1-\alpha) \cdot |\beta_j|) \right\} \tag{4.12}$$

The parameter $\alpha \in [0, 1]$ balances between the two regularizations. For $\alpha = 0$, Equation 4.12 is equal to lasso; for $\alpha = 1$, it is equal to ridge regression. For any other value of α, the elastic net combines properties of both approaches, such that coefficients can be set to 0 through the lasso penalty but can also be shrunken according to the ridge penalty.

For all the models presented above, it is common to center and scale all feature variables to mean 0 and standard deviation 1. This facilitates the interpretation of coefficients since, otherwise, features with a very large value range obtain smaller coefficients than features with a small range, even though they might be more relevant for the prediction. Furthermore, for zero-centered features, β_0 can be interpreted as the predicted response when all features are set to their mean value, i.e., 0.

Ridge regression, lasso, and elastic net are all regression methods. However, there also exist linear models for classification. Examples are *logistic regression* (which can also employ L1/L2 regularization) and linear discriminant analysis (LDA) [221]. Since these models are not applied in this thesis, we refrain from a description but refer interested readers to James et al. [222].

4.2.2 Support Vector Machines

Support vector machines (SVMs) are an algorithm for binary classification. To understand their mathematical foundation, we will first introduce a simpler algorithm known as the *maximal margin classifier* and then describe how it can be extended to a *support vector classifier*, and, finally, an SVM. We consider an input feature matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$, and a response vector $\mathbf{y} = (y_1, ..., y_N)^T$ where each entry belongs to one of two classes, either class $+1$ or class -1.

The intuition behind all classifiers presented below heavily relies on the concept of hyperplanes. In P dimensions, a hyperplane is a flat $(P-1)$-dimensional affine subspace that separates the P-dimensional space into two halves [223]. In two dimensions, a hyperplane corresponds to a line. In three dimensions, it corresponds to a plane.

In P dimensions, a hyperplane can be defined by the following set of points x_i [224]:

$$\left\{ x_i \in \mathbb{R}^P \;\middle|\; \beta_0 + \left(\sum_{j=1}^{P} \beta_j x_{ij} \right) = 0 \right\} \tag{4.13}$$

For any point x_i that is not located on this hyperplane, it holds that:

$$\beta_0 + \left(\sum_{j=1}^{P} \beta_j x_{ij} \right) > 0 \text{ or } < 0 \tag{4.14}$$

Thus, the sign of the above term denotes on which side of the hyperplane a point is located. This representation of hyperplanes can be used to derive a simple binary classifier: Assume that there exists a hyperplane that perfectly separates the samples in $\mathbf{X}$ belonging to class $+1$ from those belonging to class -1 in the P-dimensional feature space. (This is a strong assumption, which we will relax later.) If we define our hyperplane as the set of points given in Equation 4.13, we can now define a decision function as follows [223]:

$$f(x_i) = \beta_0 + \left(\sum_{j=1}^{P} \beta_j x_{ij} \right) \tag{4.15}$$

W.l.o.g., we classify sample x_i as class $+1$ if $f(x_i) > 0$ and as class -1 otherwise.
If the two classes are separable by a hyperplane in P dimensions, the number of such hyperplanes is, in fact, infinite [223]. A reasonable choice is the separating hyperplane with the largest distance from both classes. The smallest distance that any training point in $\mathbf{X}$ has from a given hyperplane is called the margin M. A large margin is typically desirable since the corresponding hyperplane should generalize well to new points and be robust to noise and small variations in the data. The separating hyperplane with the largest margin is called the *maximal margin hyperplane* and the corresponding classifier is called a *maximal margin classifier*.
Mathematically, the maximal margin classifier can be constructed via the following optimization problem [223]:

$$\max_{\beta_0, \beta_1, \ldots, \beta_P, M} \; M, \tag{4.16}$$

$$\text{such that } \sum_{j=1}^{P} \beta_j^2 = 1 \tag{4.17}$$

$$\text{and } y_i \cdot \left(\beta_0 + \left(\sum_{j=1}^{P} \beta_j \cdot x_{ij} \right) \right) \geq M \quad \forall i \in \{1, \ldots, N\} \tag{4.18}$$

The objective is to find a hyperplane that maximizes the margin M by optimizing $\beta_0, \beta_1, ..., \beta_P$. These optimized values can then be plugged into Equation 4.15 to derive a decision function $\hat{f}$. Constraint 4.17 does not constrain the hyperplane itself since $f(x_i) = 0$ and $k \cdot f(x_i) = 0$ define the same hyperplane for $k \neq 0$. However, this constraint ensures that $y_i \cdot (\beta_0 + (\sum_{j=1}^{P} \beta_j \cdot x_{ij}))$ is equal to the perpendicular distance of point x_i to the hyperplane. Consequently, Constraint 4.18 ensures that all points are located on the correct side of the hyperplane and that the distance of each point to the hyperplane is at least M.

In many real-world applications, the two considered classes may not be perfectly separable through a hyperplane in the P-dimensional space. Consequently, there is no solution for the above optimization with $M > 0$ [223]. A generalization of the maximal margin classifier that is suitable for such cases is the *support vector classifier* or *soft margin classifier* [225]. The idea is to still maximize the margin but allow some points to lie on the wrong side of the hyperplane, leading to their misclassification. Additionally, some points may be located on the correct side of the hyperplane, but their distance to the hyperplane is smaller than M.

$$\max_{\beta_0,\beta_1,...,\beta_P,\epsilon_1,...,\epsilon_N,M} \quad M, \tag{4.19}$$

$$\text{such that} \sum_{j=1}^{P} \beta_j^2 = 1 \tag{4.20}$$

$$\text{and } y_i \cdot \left(\beta_0 + \left(\sum_{j=1}^{P} \beta_j \cdot x_{ij}\right)\right) \geq M \cdot (1 - \epsilon_i) \quad \forall i \in \{1, ..., N\} \tag{4.21}$$

$$\text{and } \epsilon_i \geq 0 \quad \forall i \in \{1, ..., N\} \tag{4.22}$$

$$\text{and } \sum_{i=1}^{N} \epsilon_i < E_{\text{total}} \tag{4.23}$$

This formulation introduces N so-called *slack variables* ϵ_i with $i \in \{1, ..., N\}$. If $\epsilon_i > 0$, then point x_i violates the margin: For $\epsilon_i \in (0, 1)$, the point is located on the correct side of the hyperplane but its distance to the hyperplane is $< M$. For $\epsilon_i = 1$, the point lies directly on the hyperplane. Finally, if $\epsilon_i > 1$, the point is located on the wrong side of the hyperplane, which would lead to its misclassification. $E_{\text{total}} \geq 0$ limits the total size of the slack variables, i.e., this parameter controls the amount and severity of margin/hyperplane violations. For $E_{\text{total}} = 0$, the support vector classifier is equal to the maximal margin classifier, allowing no margin/hyperplane violations.

Interestingly, the position of the optimized hyperplane only depends on those observations that lie on the margin or that violate the margin. These points are also called the *support vectors* [225].

The support vector classifier is linear since the derived hyperplane is defined by a linear equation (cf. Equation 4.13). To model non-linear decision boundaries, we finally introduce the *support vector machine* (SVM). To better understand SVMs, we rewrite the decision function f defined in Equation 4.15: Using Lagrange multipliers to solve the optimization problem given by Equations 4.19 to 4.23, it can be shown that the optimal coefficients β_j with $j \in \{1, ..., P\}$ can be expressed as linear combination of the training data using sample-specific parameters α_i with $i \in \{1, ..., N\}$ (see [224] for details):

$$\beta_j = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_{ij} \quad \forall j \in \{1, ..., P\} \tag{4.24}$$

Consequently, the decision function in Equation 4.15 can be rewritten as follows [224]:

$$\begin{align} f(x_i) &= \beta_0 + \left(\sum_{j=1}^{P} \beta_j x_{ij} \right) \tag{4.25} \\ &= \beta_0 + \left(\sum_{j=1}^{P} \left(\sum_{i'=1}^{N} \alpha_{i'} \cdot y_{i'} \cdot x_{i'j} \right) \cdot x_{ij} \right) \tag{4.26} \\ &= \beta_0 + \left(\sum_{i'=1}^{N} \alpha_{i'} \cdot y_{i'} \cdot \left(\sum_{j=1}^{P} x_{ij} \cdot x_{i'j} \right) \right) \tag{4.27} \\ &= \beta_0 + \left(\sum_{i'=1}^{N} \alpha_{i'} \cdot y_{i'} \cdot \langle x_i, x_{i'} \rangle \right) \tag{4.28} \end{align}$$

Here, $\langle x_i, x_{i'} \rangle$ denotes the dot product:

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{P} x_{ij} \cdot x_{i'j} \tag{4.29}$$

Based on this definition of f, we can introduce the concept of SVMs: The idea behind SVMs is to overcome the problem of non-linearly separable classes by transforming the original P-dimensional data points into a higher dimension and to define a separating hyperplane in this higher-dimensional space. Thus, each original data point x_i is replaced with its transformation $h(x_i)$ [226]:

$$f(x_i) = \beta_0 + \sum_{i'=1}^{N} \alpha_{i'} \cdot y_{i'} \cdot \langle h(x_i), h(x_{i'}) \rangle \tag{4.30}$$

Typically, the transformation is not explicitly computed. Instead, a so-called kernel function $\mathcal{K}(x_i, x_{i'})$ is used [226]:

$$f(x_i) = \beta_0 + \sum_{i'=1}^{N} \alpha_{i'} \cdot y_{i'} \cdot \mathcal{K}(x_i, x_{i'}) \tag{4.31}$$

A kernel function computes the dot product of two points in the transformed space without explicitly calculating their transformation. Thus, using kernel functions is computationally efficient. Common examples are *polynomial kernels* of degree d [226], e.g.,

$$\mathcal{K}(x_i, x_{i'}) = (1 + \langle x_i, x_{i'} \rangle)^d \tag{4.32}$$

and the *radial basis kernel* [226]

$$\mathcal{K}(x_i, x_{i'}) = exp(-\gamma \cdot ||x_i - x_{i'}||^2) \tag{4.33}$$

$$\text{with} \quad ||x_i - x_{i'}|| = \sqrt{\sum_{j=1}^{P} (x_{ij} - x_{i'j})^2} \quad . \tag{4.34}$$

The parameters for β_0, and $\alpha_1, ..., \alpha_N$ can be optimized using Lagrange multipliers and plugged into Equation 4.31 to obtain the decision function $\hat{f}$ [226]. As described above, a sample x_i can then be classified according to the sign of $\hat{f}(x_i)$.

While SVMs cannot predict class probabilities directly, class probabilities can be obtained using a method called *Platt scaling* [227]. Briefly summarized, Platt scaling uses logistic regression to fit a logistic function with parameters A and B to the outputs of $\hat{f}$. The function yields values in $(0, 1)$ which can be interpreted as the probability of a sample belonging to class +1 [227]:

$$Pr(y_i = +1|x_i) = \frac{1}{1 + exp(A \cdot f(x_i) + B)} \tag{4.35}$$

The class probability for class -1 is simply $1 - Pr(y_i = +1|x_i)$. Additionally, SVMs can also be extended to perform multi-class classification [228] and regression [229].

4.2.3 K-Nearest Neighbors

A simple ML model that can be applied for both regression and classification is the K-nearest neighbors (KNN) model. The concept behind KNN was initially developed by Fix and Hodges [230, 231] and later expanded by Cover and Hart [232]. For a previously unseen sample x_j, KNN first identifies a set $\mathcal{N}_j$ consisting of the K training samples that are closest to the new sample in the input feature space. These samples represent the *nearest neighbors* of x_j. To determine $\mathcal{N}_j$, different distance measures d can be applied. For continuous features, the Euclidean distance is typically used, which is defined as the Euclidean (L2) norm of the difference between the feature vectors for two samples x_i and x_j (cf. Equation 4.34):

$$d(x_i, x_j) = ||x_i - x_j|| \tag{4.36}$$

The K training samples, for which the distance to x_j is smallest, are added to $\mathcal{N}_j$. Before computing Equation 4.36, all features should be scaled to the same range. Otherwise, features with larger scale impact the distances more than those with a smaller scale.
Finally, for regression, the response for x_j is predicted as the (weighted) average over all samples in $\mathcal{N}_j$ [233]:

$$\hat{f}(x_j) = \sum_{x_i \in \mathcal{N}_j} w_i \cdot y_i \tag{4.37}$$

For classification using a set of classes C, a (weighted) majority vote is used [207]:

$$\hat{f}(x_j) = \underset{c \in C}{\operatorname{argmax}} \left\{ \sum_{x_i \in \mathcal{N}_j} w_i \cdot I(y_i = c) \right\} \tag{4.38}$$

Here, $I(y_i = c)$ denotes an indicator function that is 1 if y_i is equal to class c and 0 otherwise. In both equations, the sample weights w_i can be set to $\frac{1}{N}$ to weight all samples equally.
Instead of using a KNN classifier to predict just a single class for x_j, we can also predict a class probability for each class $c \in C$:

$$\hat{Pr}(c|x_j) = \frac{\sum_{x_i \in \mathcal{N}_j} w_i \cdot I(y_i = c)}{\sum_{x_i \in \mathcal{N}_j} w_i} \tag{4.39}$$

Intuitively, $\hat{Pr}(c|x_j)$ is the weighted fraction of neighbors in $\mathcal{N}_j$ that belong to class c.

4.2.4 Tree-Based Methods

In this thesis, we employ two types of tree-based methods, namely random forests and boosting trees. What distinguishes these methods from the linear regression and KNN models discussed in the previous sections is that they are so-called *ensemble models*: Ensemble models require training multiple (often simple and less powerful) models, known as *weak learners*, which are then combined into one (potentially very powerful) model [234].
The weak learners in tree-based methods are so-called *decision trees*. These models are remarkable since their structure mirrors human decision-making through a sequence of if-then-else decisions. An exemplary binary decision tree is shown in Figure 4.1A. It has a top-down structure consisting of nodes representing binary decisions on the features (e.g., 'Is the value of feature $X_1 < 5$?'). Each internal node is followed by two branches that denote the possible decisions, i.e., *true* (left branch) or *false* (right branch). After several decisions, a leaf node is reached, where a prediction is made.

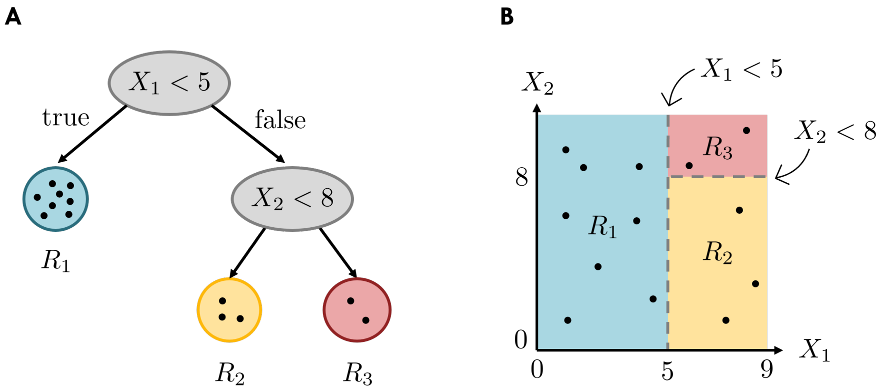


FIGURE 4.1: Exemplary Decision Tree. Subfigure A depicts an example of a decision tree consisting of two internal nodes (gray) and three leaf nodes (blue, yellow, red). Each internal node represents a binary decision on the features, and the two child notes correspond to the outcome (i.e., *true* or *false*). Based on these decisions, each sample (black dots) falls into exactly one leaf node. Subfigure B depicts the feature space defined by features X_1 and X_2. It shows how the binary decisions of the tree partition the space (and the samples therein) into three non-overlapping regions R_1, R_2, R_3.

In the following, we first discuss how to build such a decision tree and how to make predictions with it. Afterward, we introduce random forests and boosting trees, which implement the idea of tree ensembles in two distinct manners. Especially random forests play a special role in this thesis: they are the basis for our prediction approach SAURON-RF (cf. Chapters 6 and 7), and we found them to be well suited for the sensitivity prediction of monotherapies (cf. Chapter 5) and combination therapies (cf. Chapter 8).

4.2.4.1 Decision Trees

As discussed above, a single decision tree can be interpreted as a sequence of if-then-else decision rules on the available features. These rules partition the feature space into L distinct, non-overlapping regions $R_1, ..., R_L$, defined by the leaf nodes of the tree, where each data sample falls into exactly one region [235] (cf. Figure 4.1B). Predictions are constant for each region R_l with $l \in \{1, ..., L\}$ and are calculated as the average (regression) or majority vote (classification) over the responses of all training samples in region R_l [235]. Consequently, the aim is to determine these regions such that predictions for the training samples are as accurate as possible. However, it is computationally infeasible to investigate all possible partitions of the feature space to find the best partition. Consequently, decision trees are built using a greedy heuristic known as *recursive binary splitting* [235]. The approach starts with the entire feature space as one region, represented by the root node of the tree, which initially has no child nodes. Starting at the root, each node v in the tree is recursively split into a left child node v_l and a right child node v_r. Intuitively, each splitting partitions one region into two subregions. The aim of splitting nodes is to improve predictions by choosing child nodes v_l and v_r with a reduced error compared to the parent node v. To this end, the predicted response for any sample reaching node v for regression is given by:

$$\hat{y}^v = \sum_{i \in \delta(v)} w_i^v \cdot y_i \tag{4.40}$$

Here, $\delta(v)$ describes the set of training samples in node v, and w_i^v is a sample-specific weight, which is commonly set to $\frac{1}{|\delta(v)|}$ for all training samples, such that Equation 4.40 is an ordinary average. The node-specific prediction error of v for regression problems can be computed using the (weighted) mean squared error (MSE) over all training samples in v:

$$\text{MSE}(v) = \sum_{i \in \delta(v)} w_i^v \cdot (y_i - \hat{y}^v)^2 \tag{4.41}$$

To perform a splitting of node v into two child nodes v_l and v_r, a feature X_j and a cutpoint s are determined such that:

$$\delta(v_l) = \{i \in \delta(v) \,|\, x_{ij} < s\} \quad \text{and} \quad \delta(v_r) = \{i \in \delta(v) \,|\, x_{ij} \geq s\} \tag{4.42}$$

For regression, the feature and cutpoint are chosen to maximize:

$$w_{an}(v) \cdot (\text{MSE}(v) - w_{ch}(v_r) \cdot \text{MSE}(v_r) - w_{ch}(v_l) \cdot \text{MSE}(v_l)) \tag{4.43}$$

Here, w_{an} and w_{ch} are node-specific weights for the ancestor and child nodes, respectively, which are typically set to $w_{an}(v) = \frac{|\delta(v)|}{N}$, $w_{ch}(v_r) = \frac{|\delta(v_r)|}{|\delta(v)|}$, and $w_{ch}(v_l) = \frac{|\delta(v_l)|}{|\delta(v)|}$. This procedure is repeated recursively by splitting the newly generated nodes until some stopping criterion is fulfilled. Typical stopping criteria include reaching a minimal number of samples in a node or a maximum number of (successive) splits in a tree. Additionally, a node is not split if none of the potential splits would result in an error improvement.

Finally, the prediction for a new sample is computed by passing the sample through the tree to determine its leaf node, i.e., its region, and returning the region-specific prediction. For a sample x reaching leaf node μ, the prediction is given by (cf. Equation 4.40):

$$\hat{f}(x) = \hat{y}^\mu = \sum_{i \in \delta(\mu)} w_i^\mu \cdot y_i \tag{4.44}$$

While the above descriptions focus on regression, decision trees can also be used to perform a classification using a set of K classes $C = \{c_1, ..., c_K\}$ [235]. Since the general procedure is the same for regression and classification, we will only provide the required equations in the following.

The predicted response for any sample reaching node v is given by a weighted majority vote over the responses in v:

$$\underset{c \in C}{\operatorname{argmax}} \left\{ \sum_{i \in \delta(v)} w_i^v \cdot I(y_i = c) \right\} \tag{4.45}$$

The indicator function $I(y_i = c)$ is 1 if y_i is equal to c and 0 otherwise. The remaining error in node v is measured using the Gini impurity, also known as the Gini index:

$$\text{Gini}(v) = \sum_{c \in C} \hat{p}_c^v \cdot (1 - \hat{p}_c^v) \tag{4.46}$$

$$\text{with} \quad \hat{p}_c^v = \frac{\sum_{i \in \delta(v)} w_i^v \cdot I(y_i = c)}{\sum_{i \in \delta(v)} w_i^v} \tag{4.47}$$

Intuitively, $\hat{p}_c^v$ denotes the weighted fraction of samples belonging to class c in node v. If all samples are weighted equally using $w_i^v = \frac{1}{|\delta(v)|}$, then $\hat{p}_c^v = \frac{\sum_{i \in \delta(v)} I(y_i = c)}{|\delta(v)|}$. Consequently, $\hat{p}_c^v$ can be interpreted as a predicted class probability for class c in node v since

$$\forall v : \sum_{c \in C} \hat{p}_c^v = 1 \quad . \tag{4.48}$$

A node v is split into two child nodes v_l and v_r by selecting a feature and cutpoint (cf. Equation 4.42) such that the resulting nodes maximize:

$$w_{an}(v) \cdot (\text{Gini}(v) - w_{ch}(v_r) \cdot \text{Gini}(v_r) - w_{ch}(v_l) \cdot \text{Gini}(v_l)) \tag{4.49}$$

Finally, for a sample x reaching leaf node μ, the classification prediction is determined by majority vote:

$$\hat{f}(x) = \underset{c \in C}{\operatorname{argmax}} \, \{\hat{p}_c^\mu\} = \underset{c \in C}{\operatorname{argmax}} \left\{ \sum_{i \in \delta(\mu)} w_i^\mu \cdot I(y_i = c) \right\} \tag{4.50}$$

Here, $\hat{p}_c^\mu$ is equal to the class probability of sample x_j belonging to class c:

$$\hat{Pr}(c|x_j) = \hat{p}_c^\mu \tag{4.51}$$

4.2.4.2 Random Forests

Random forests are a regression and classification algorithm developed by Breiman that comprises an ensemble of B decision trees [236]. For a given sample, each tree makes one response prediction, and the predictions over all B trees are aggregated by averaging for regression or majority vote for classification. The development of random forests was motivated by the fact that individual decision trees suffer from high variance, i.e., trees tend to differ considerably when trained on different datasets [234]. This variance can be reduced by aggregating predictions over multiple trees that are trained for the same task but de-correlated such that the forest does not consist of B identical trees (see [237] for details). The de-correlation of trees is achieved through two measures:

1. Instead of training each tree on the same data, a fixed number of samples from the training set, typically N, is drawn randomly with replacement and used to train each tree. This procedure is also known as *bootstrapping*. Since samples are drawn with replacement, the same sample may occur multiple times in the bootstrapped set for one tree and it may be absent in others. For trees in a random forest, the definition of $\delta(v)$, we gave in Section 4.2.4.1 for decision trees, has to be adapted slightly: instead of denoting the set of all training samples reaching node v, it denotes the set of all bootstrapped samples for the current tree reaching node v.

2. Instead of considering all P available features when splitting a node, only a randomly selected subset of $m < P$ unique features is considered. This is also known as *feature bagging*. Commonly, m is set to $\sqrt{P}$ for regression and $\frac{P}{3}$ for classification [238].

After building B trees, each tree can make a regression/classification prediction: The prediction of tree b for a sample x is denoted as $\hat{f}_b(x)$ and is computed as denoted in Equations 4.44 and 4.50 for regression and classification, respectively. Finally, the tree-specific predictions are aggregated by averaging for regression

$$\hat{f}(x) = \sum_{b=1}^{B} w_b(x) \cdot \hat{f}_b(x) \tag{4.52}$$

or by majority vote for classification:

$$\hat{f}(x) = \operatorname*{argmax}_{c \in C} \left\{ \sum_{b=1}^{B} w_b(x) \cdot I(\hat{f}_b(x) = c) \right\} \tag{4.53}$$

In both equations, $w_b(x)$ is a tree-specific weight denoting the influence of each tree on the predictions for sample x. Conventionally, all trees are weighted equally, i.e., each weight is set to $\frac{1}{B}$. In Chapter 6, we show how weighting trees differently, depending on the considered sample x, can substantially improve predictions.

Just like decision trees, random forests can also be used to predict class probabilities. One possibility is to compute class probabilities as the (weighted) fraction of trees that voted for a specific class c:

$$\hat{Pr}(c|x) = \frac{\sum_{b=1}^{B} w_b(x) \cdot I(\hat{f}_b(x) = c)}{\sum_{b=1}^{B} w_b(x)} \tag{4.54}$$

4.2.4.3 Boosting Trees

Similar to random forests, boosting tree models comprise an ensemble of decision trees. However, boosting trees are built in an iterative rather than a parallel fashion, as in the random forest approach. In each iteration, a new tree is added to the ensemble. This tree is not trained to learn the response $\mathbf{y}$ but instead the residuals $\mathbf{r} = (r_1, ..., r_N)^T$. The residuals are defined as the difference between the actual response and the prediction of the current model:

$$r_i = y_i - \hat{f}(x_i) \quad \forall i \in \{1, ..., N\} \tag{4.55}$$

Consequently, each new tree aims to eliminate the errors of the model consisting of all previously built trees. The boosting tree algorithm for regression can be described as follows [234]:

1. Set the initial model to $\hat{f}(x) = 0$ and the residuals to $r_i = y_i$ for all training samples $i \in \{1, ..., N\}$.
2. For $b = 1, ..., B$ repeat the following steps:
   - Fit a tree with $d \in \mathbb{N}^+$ splits to the training data $(\mathbf{X}, \mathbf{r})$ resulting in the model $\hat{f}^b$.
   - Update the ensemble model:

     $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \cdot \hat{f}^b(x) \tag{4.56}$$

     Here, $\lambda \in (0, 1]$ is a shrinkage parameter, also known as the learning rate of the model.
   - Update the residuals:

     $$r_i \leftarrow r_i - \lambda \cdot \hat{f}^b(x_i) \quad \forall i \in \{1, ..., N\} \tag{4.57}$$

3. The final boosting model is then given by:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \cdot \hat{f}^b(x) \tag{4.58}$$

While increasing the number of trees B does typically not lead to overfitting for random forests, boosting trees can suffer from overfitting when B becomes too large. Similarly, the parameter d controlling the size of each tree is typically small and can even be set to 1. Also, note that the trees trained in Step 2 can use sample weights as described in Section 4.2.4.1.

The algorithm described above assumes continuous values for the actual and predicted responses. Consequently, it is suited for regression. For classification, a common boosting method often used with trees is AdaBoost (Adaptive Boosting). AdaBoost was originally developed by Freund and Shapire for binary classification [239]; however, a multi-class extension has been developed by Hastie et al. [240]. Briefly summarized, AdaBoost is also based on the idea of training several weak learners (here: decision trees) in an iterative fashion. Unlike the algorithm presented above that iteratively trains trees on residuals, AdaBoost trees are all trained on the same class vector but with different sample weights: The model $\hat{f}^b$ trained in each iteration b employs sample weights that increase the importance of samples that were misclassified by the ensemble model trained in the previous iteration. The ensemble prediction is a weighted majority vote over all trees, where more accurate learners obtain a higher weight.

4.2.5 Neural Networks

Neural networks are prediction models that are loosely modeled after mechanisms of neural signal transmission in the human brain [241]. They are widely popular for various ML tasks and the recent breakthroughs in generative AI can largely be attributed to the development of highly advanced neural networks, such as the large language model ChatGPT [195] or the image generator DALL·E [196]. Hence, it comes as no surprise that neural networks are likewise a popular choice for drug sensitivity prediction [25, 242, 243] and we also consider them in our studies (cf. Chapters 5 to 8).
A variety of neural network architectures exist, which are often tailored to specific application cases: Convolutional neural networks, e.g., are commonly used for image analysis, while recurrent neural networks can model sequential data such as text or speech [241]. In this thesis, we limit our descriptions to feed-forward neural networks and the fundamentals of their application for supervised learning, as these are the only types of neural networks employed in our work. For a detailed description of various types of neural networks, we recommend the books by Goodfellow et al. [241] and James et al. [171].

4.2.5.1 Network Structure

A typical feed-forward neural network consists of multiple consecutive layers of nodes, which are connected through weighted edges. The first network layer is known as the input layer, where each node corresponds to one predictor variable X_j with $j \in \{1, ..., P\}$. The input layer is followed by one or several hidden layers of nodes, which are then followed by an output layer, where each node corresponds to a response variable. For simplicity, we limit the following descriptions to neural networks with only one output node, predicting a single response variable Y. An example of a neural network is depicted in Figure 4.2.

The term *feed-forward* is used to describe neural networks where the flow of information is unidirectional. This means that edges between nodes are always directed to point from earlier layers in the network to later layers and never back to previous layers [244]. The simplest form of such a network is known as a *fully-connected* feed-forward network, where all nodes from one layer are connected to all nodes of the consecutive layer.

Neural networks make predictions by passing the predictor values $x_{i1}, ..., x_{iP}$ of a sample i to the nodes of the input layer. Each node then processes the received information and passes the resulting value on to nodes in the consecutive layers via weighted edges. This process is repeated in each layer until the output layer is reached, where the final response for sample i is predicted.

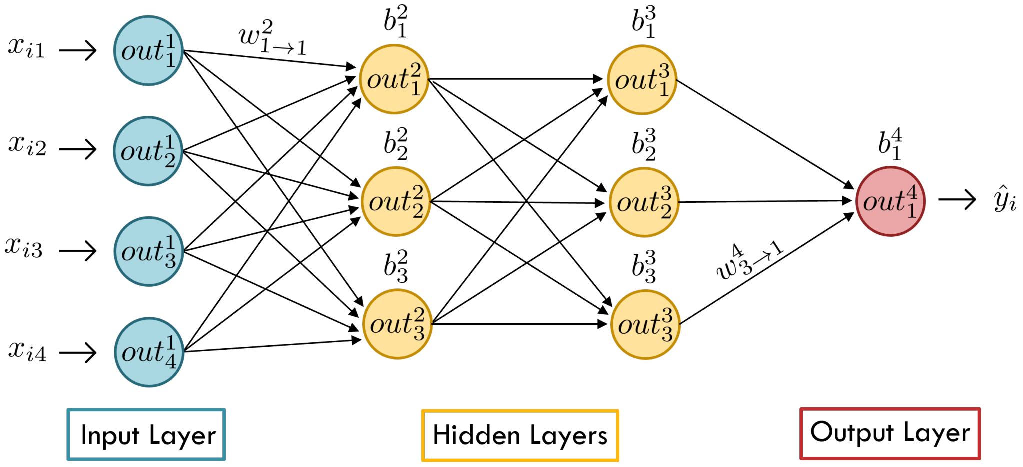


FIGURE 4.2: Example neural network. This figure depicts an exemplary neural network. The input layer, which consists of four nodes, each corresponding to one predictor variable, is shown in blue. The model inputs for a sample x_i are shown on the left. The model's two hidden layers are depicted in yellow, and the output layer, consisting of a single node predicting the response $\hat{y}_i$, is shown in red. Furthermore, each node is labeled with its output *out*. Node biases b in the hidden and output layers are depicted on top of the corresponding nodes. Additionally, two exemplary edge weights w are shown.

To learn the relationship between input and output variables, neural networks use two kinds of trainable parameters, namely edge-specific weights and node-specific biases. To describe the information flow through the network using these parameters, we need to introduce several notations (cf. Figure 4.2): let L denote the total number of network layers, and N_l the number of nodes in layer l. Note that for P predictor variables and one response variable, $N_1 = P$ and $N_L = 1$. The weight of an edge connecting node u in layer $l-1$ to node v in layer l is then given by $w^l_{u\to v}$, and the bias of node v in layer l is given by b^l_v.

Based on these definitions, the flow of information through the network can be summarized based on the output signal of node v in layer l as follows:

$$out^l_v = \begin{cases} x_{iv}, & \text{if } l = 1 \\ \sigma((\sum_{u=1}^{N_{l-1}} w^l_{u\to v} \cdot out^{l-1}_u) + b^l_v), & \text{else} \end{cases} \tag{4.59}$$

The first case in Equation 4.59 describes the input layer ($l = 1$): Here, the output of each node simply corresponds to the predictor values of sample i. For all consecutive layers ($l > 1$), we first collect the incoming signal of all nodes in the previous layer (out^{l-1}_u), weighted by the respective edge weights $w^l_{u\to v}$. Additionally, the node-specific bias b^l_v is added. Finally, an activation function σ is applied to obtain the final node output. The activation function serves two purposes: (1) It is typically a non-linear function, allowing the network to model non-linear relationships between predictors and responses. (2) The activation function typically has a switch-like behavior, such that it only passes a (strong) signal forward if its inputs are sufficiently large. Common examples are the sigmoid-function ($\sigma(x) = 1/(1+e^{-x})$) and ReLU ($\sigma(x) = max(0,x)$) [245, 246]. This switch-like behavior was motivated by signal transmission in biological neurons, where neurons only transmit a signal if their incoming stimulus exceeds a certain threshold [247]. In artificial neural networks, this behavior can, e.g., be useful to filter out noise or minute and, thus, potentially irrelevant signals. In practice, the activation function is commonly the same for all nodes in the hidden layers. In the output layer, it is typically chosen to ensure that network outputs are in the desired range. For example, ReLU yields predictions ≥ 0, while the sigmoid yields predictions in $(0, 1)$.

Finally, the prediction for sample x_i is directly given by the node output in the last layer:

$$\hat{y}_i = out^l_1 \tag{4.60}$$

4.2.5.2 Network Training

Neural networks are trained using an iterative procedure that can be divided into the following four steps:

1. Parameter Initialization:
   The network parameters (i.e., weights and biases) are initialized to some starting values. Common techniques include the He [248] and Glorot [249] initializations, which sample parameters from a normal or a uniform distribution, respectively.

2. Network Application:
   The network is applied to the training data, resulting in a prediction for each training sample.

3. Error Calculation:
   The prediction error is determined by using a loss function L such as the MSE (cf. Equation 4.65) that compares the actual responses of the training data to the predicted responses. The error function may also contain sample weight to increase the impact of certain samples on model training (cf. Equation 4.41).

4. Parameter Adaptation:
   The weights and biases of the network are adapted such that the error is reduced. Typically, parameter updates are computed using a technique called *gradient descent* [250]. The idea is to compute the gradient of the loss function, i.e., the partial derivative of the loss w.r.t. each network parameter. Next, parameters are updated using the negative gradient, which points in the direction of the steepest descent of the loss function:

$$w^l_{u\to v} \leftarrow w^l_{u\to v} - \lambda \frac{\delta L}{\delta w^l_{u\to v}} \tag{4.61}$$

$$b^l_v \leftarrow b^l_v - \lambda \frac{\delta L}{\delta b^l_v} \tag{4.62}$$

   The parameter $\lambda > 0$ is known as the *learning rate*. It controls how much the gradient computed in a single iteration changes the weights and biases.

Steps 2 to 4 are repeated until some stopping criterion is fulfilled, e.g., until a certain number of iterations is reached or until the error converges [241].

Since neural networks can have thousands of parameters, the computation of the gradient required in Step 4 is computationally expensive. Consequently, the *backpropagation* algorithm has been developed by Rumelhart et al. [251], which employs the chain rule of calculus to recursively propagate the loss from latter to earlier network layers, thereby speeding up gradient computations significantly. Today, several variations and extensions of the gradient descent and backpropagation algorithms exist [250].

4.2.5.3 Classification

From Equation 4.59, it is apparent that neural networks can only predict continuous values. However, they can still be used for classification: For binary classification, the two classes can, e.g., be encoded as responses of 0 and 1. After applying a sigmoid activation function in the last network layer, the predictions can be interpreted as the probability of the input sample belonging to class 1. Class predictions can then be derived by using a threshold, e.g., 0.5, and classifying all samples with predicted probability greater than this threshold as class 1.

Neural networks can also be used for multi-class classification with classes $C = \{c_1, ..., c_K\}$: The idea is to introduce one output node for each class and apply the softmax function as activation in the output layer [245]. Let in_k^L denote the signal processed by node k in layer L before applying the activation function, i.e., $out_k^L = \sigma(in_k^L)$, then the softmax function is given by [252]:

$$\sigma(in_k^L) = \frac{e^{in_k^L}}{\sum_{k'=1}^{K} e^{in_{k'}^L}} \tag{4.63}$$

Consequently, the prediction of each output node k can be interpreted as the predicted class probability for class c_k, and a sample x_i is finally predicted to belong to the class with the largest probability, i.e., $\hat{y}_i = \text{argmax}_{c_k \in C} \left\{ out_k^L \right\}$.

A common loss function for classification is the cross entropy [253]:

$$\text{CE} = -\sum_{i=1}^{N} \sum_{k=1}^{K} w_i \cdot I(y_i = c_k) \cdot log(\hat{y}_i) \tag{4.64}$$

Here, $I(y_i = c_k)$ is 1 if the actual class of sample x_i is c_k and 0 otherwise, and w_i is a sample-specific weight.

4.2.6 Hyperparameter Tuning

In the previous sections, we discussed several ML algorithms in detail. Each algorithm assumes a certain model structure and estimates model parameters using the training data such that prediction quality is optimized. Examples of the learned parameters are the coefficients for linear regression models (cf. Section 4.2.1) or the weights and biases in neural networks (cf. Section 4.2.5). However, each algorithm also comprises so-called *hyperparameters* that have to be set prior to model training. Examples include the choice of α in an elastic net (cf. Equation 4.12), the number of trees in a random forest, or the number of hidden layers in a neural network. A simpler model with reasonable hyperparameter choices can outperform a sophisticated, complex model with poorly chosen hyperparameters. However, it is typically unknown beforehand which hyperparameters result in the best performance for a given problem and dataset. Consequently, hyperparameters are commonly tuned through k-fold cross-validation (cf. Section 4.1.1) on the training data: For each hyperparameter combination, k models are trained using the CV training folds. The hyperparameter combination resulting in the best error averaged across all k test folds is then used to train a final model on the entire training data, which can then be evaluated using the test data.

4.3 Requirements of Trustworthy Machine Learning Models

As stated in the introductory section of this chapter, application of AI in sensitive fields such as health care requires special diligence to avoid harmful consequences. Recognizing the need for system regulation, the European Union passed the EU AI Act in 2024 [254]. Therein, regulations are described that should guarantee that the practical application of AI systems is safe, lawful, and ethical. For example, AI systems should not discriminate against certain social groups, and security mechanisms should prevent (accidental) misuse of the system and its underlying data. These requirements can be summarized under the term *trustworthiness*.

In the following sections, we discuss several properties trustworthy AI should fulfill from an ML perspective. These include a good prediction performance, the reliability of individual predictions, and a high interpretability such that a model's decision-making can be understood by humans. Additionally, an AI system should only be considered trustworthy if it is tailored to the specific task at hand, which will be discussed in the last section of this chapter. As mentioned above, there exist further requirements that trustworthy AI systems should fulfill [26, 255, 256], such as security and safety mechanisms that protect the system against accidental or intentional misuse, which are, however, out of the scope of this thesis.

4.3.1 Performance

Clearly, an ML model can only be trusted if its predictions for the desired task are sufficiently accurate. Consequently, there exist various error measures to quantify the predictive capabilities of both regression and classification models during training and testing, i.e., when a ground truth is available. In supervised learning, such measures are also explicitly incorporated into model training to optimize prediction correctness (cf.

Section 4.1.1). In the following, we present several error measures that we employed for our analyses in Chapters 5 to 9. For a comprehensive summary of the most common error measures, please refer to Naser and Alavi [257].

For regression, a widely used error measure is the mean squared error MSE $\in [0, \infty)$: Let $\mathbf{y} = (y_1, ..., y_N)^T$ and $\hat{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_N)^T$ denote the N-dimensional vectors containing the actual and predicted responses for each sample, respectively, then the MSE is defined as:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4.65}$$

The more model predictions deviate from the actual responses, the more the MSE increases. Common variations of the MSE are the root mean squared error (RMSE) defined as $\sqrt{MSE}$ and the mean absolute error (MAE), where the squared deviation is replaced by the absolute deviation.

Other common error measures for regression are correlation coefficients that measure the presence of certain trends between the actual and predicted responses. Most correlation coefficients have the benefit of always being in the range $[-1, 1]$. This makes them easier to compare across experiments than the (R)MSE and MAE, which have no upper bound. For example, the Pearson correlation coefficient PCC $\in [-1, 1]$ can be used to measure the linear dependency between the actual and predicted responses:

$$PCC = \frac{\sum_{i=1}^{N} (y_i - \bar{y}) \cdot (\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2 \sum_{i=1}^{N} (\hat{y}_i - \bar{\hat{y}})^2}} \tag{4.66}$$

$$\text{with} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{4.67}$$

$$\text{and} \quad \bar{\hat{y}} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i \tag{4.68}$$

For a perfect positive linear correlation, $PCC = 1$, and for a perfect negative linear correlation, PCC $= -1$. A value of PCC $= 0$ indicates no linear dependency between the two variables. To measure non-linear dependencies, alternative coefficients such as the Spearman correlation coefficient (SCC) [258] or Kendall's τ coefficient [259] can be used. SCC, for example, measures a monotonic relationship between two variables and is defined as the PCC between the rank values of $\mathbf{y}$ and $\hat{\mathbf{y}}$.

For classification tasks, there likewise exists a plethora of error measures. Here, we limit our descriptions to the binary classification case, where each sample belongs to one of two classes, a *positive* class or a *negative* class. Consequently, each prediction of a classification model belongs to exactly one of four cases:

- True Positive (TP): a positive sample was correctly predicted as positive
- True Negative (TN): a negative sample was correctly predicted as negative
- False Positive (FP): a negative sample was falsely predicted as positive
- False Negative (FN): a positive sample was falsely predicted as negative

Based on these definitions, different error measures can be defined. The most straightforward measure is the prediction *accuracy* $\in [0,1]$, which measures the fraction of correctly classified samples among all samples:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.69}$$

Analogously, the measures *sensitivity* $\in [0,1]$ and *specificity* $\in [0,1]$ measure the fraction of correctly classified positive and negative samples among all positive and negative samples, respectively:

$$sensitivity = \frac{TP}{TP + FN} \tag{4.70}$$

$$specificity = \frac{TN}{TN + FP} \tag{4.71}$$

A drawback of the accuracy measure is that it can be misleading for highly imbalanced datasets: Consider a dataset where 90% of samples are negative and only 10% are positive. A classifier that predicts any sample to be negative will still achieve an accuracy of 0.9 despite having no discriminatory power at all. This phenomenon would, however, become apparent when inspecting the model's sensitivity, which is 0. Consequently, model accuracy should not be considered in isolation. Even for balanced datasets, sensitivity and specificity can provide valuable insight into whether one of the classes is more difficult to predict correctly. A measure that combines both sensitivity and specificity is the *balanced accuracy*, which is defined as the mean of both values [260].
Another measure that gives equal importance to the accurate prediction of positive and negative samples is Matthew's correlation coefficient MCC $\in [-1,1]$ [261]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{4.72}$$

MCC is 1 for a perfect classification, whereas a classifier that always predicts the wrong class results in MCC $= -1$.

As discussed in Section 4.2, many classification algorithms estimate some form of class probability for each class, and the final prediction corresponds to the class with the highest probability. For a binary classification with positive class c^{pos} and negative class c^{neg}, we can denote the predicted probability of class c^{pos} for sample x_i as $\hat{y}_i^{\text{pos}}$ and the predicted probability of class c^{neg} as $\hat{y}_i^{\text{neg}} = 1 - \hat{y}_i^{\text{pos}}$. Next, a threshold t can be applied to $\hat{y}_i^{\text{pos}}$ to classify x_i as follows:

$$\hat{y}_i = \begin{cases} c^{\text{pos}} \;\; (\textit{positive}), & \text{if } \hat{y}_i^{\text{pos}} \geq t \\ c^{\text{neg}} \;\; (\textit{negative}), & \text{else} \end{cases} \tag{4.73}$$

Using the conventional threshold of $t = 0.5$, a sample is always assigned to the class with the highest probability (or to the positive class if $\hat{y}_i^{\text{pos}} = \hat{y}_i^{\text{neg}} = 0.5$). However, t can be varied freely between 0 and 1.

The performance of a classifier for varying t in terms of sensitivity and specificity can be visualized using a *receiver operating characteristic* (ROC) curve, as depicted in Figure 4.3 [221]. This visualization shows that lowering the threshold may increase sensitivity (but decrease specificity), while increasing the threshold has the opposite effect. Such trade-offs between sensitivity and specificity can be desirable, depending on the model application. For personalized treatment recommendations, e.g., we would want to avoid falsely classifying ineffective treatments as effective to avoid treating a patient with an ineffective drug. To achieve this, we might be willing to let our model fail to detect some potentially effective treatments.

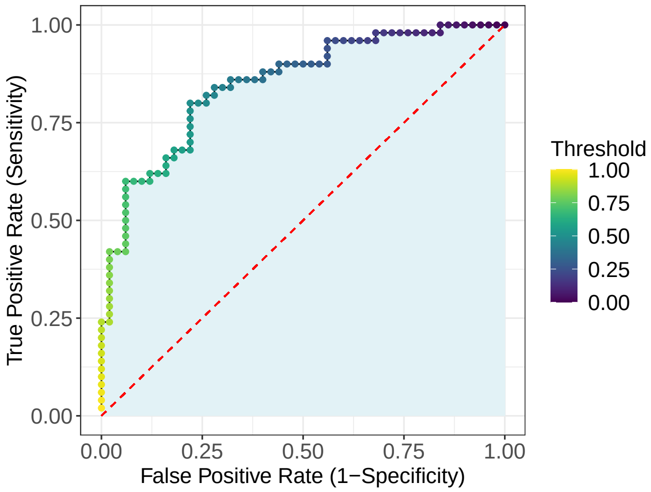


FIGURE 4.3: Receiver operating characteristic (ROC) curve. This figure depicts a ROC curve showing the true positive rate (i.e., sensitivity) and false positive rate (i.e., 1 - specificity) of an exemplary classifier for different thresholds t (cf. Equation 4.73). The area under the ROC curve (AUC) of this classifier is highlighted in light blue. The red dashed line shows the ROC curve for a random classifier with AUC = 0.5.

A measure that condenses the information of the entire ROC curve into a single value is the AUC, i.e., the *area under the ROC curve* [221], highlighted in light blue in Figure 4.3. A classifier, where the probability of the correct class exceeds the probability of the incorrect class for every sample, has an AUC of 1 [262]. In contrast, a random classifier that randomly samples class probabilities from a uniform distribution in $[0, 1]$ has an AUC of 0.5 [263], even for imbalanced datasets: For any threshold t, the true positive rate (sensitivity) and false positive rate (1 - specificity) are identical and equal to $1 - t$ (cf. red dashed line in Figure 4.3).

4.3.2 Reliability

Performance measures, as described in the previous section, can be calculated as long as a ground truth is known (i.e., during training and testing) to estimate a model's performance. However, in real-world applications of a model, the goal is to make predictions for unseen samples where no known response is available and, thus, no performance measure can be computed. Consequently, there is a need to estimate the degree of trust we have in individual predictions for previously unseen samples, which is commonly referred to as *reliability* [27–29].

One way to achieve reliability is through *uncertainty quantification* [264, 265], which focuses on estimating the predictive uncertainty associated with an individual prediction. Intuitively, this can be compared to a medical doctor, who is more confident in their diagnosis of certain patients compared to others.

Two types of uncertainty that can be distinguished are *aleatoric* and *epistemic* uncertainty [266]. Aleatoric (or statistical) uncertainty refers to the uncertainty within the data, which is caused by the inherent randomness in data generation, including noise [266]. In contrast, epistemic (or systematic) uncertainty stems from our lack of knowledge about how to best model the relationship between predictors and response [266], which includes the choice of model class, the way model parameters are estimated, and the amount of available data.

Uncertainty is directly linked to prediction error. This becomes apparent when looking back at the bias-variance trade-off, which decomposes the expected squared error of an ML model into model bias, model variance, and irreducible error (cf. Equation 4.5). Here, aleatoric uncertainty corresponds to the irreducible error [265]. In contrast, the epistemic uncertainty corresponds to the reducible error, i.e., the bias and variance of the model [265]. More specifically, the type of epistemic uncertainty that corresponds to the model bias is known as *model uncertainty* and describes the uncertainty we introduce by choosing a certain model class with certain hypotheses, e.g., choosing a model that can only model linear relationships [265]. The epistemic uncertainty that corresponds to the

model variance is known as *approximation uncertainty*, which comprises the uncertainty related to our estimates of the model parameters using the training data [265].
Knowledge about the uncertainty associated with individual predictions can greatly increase our trust in a model. Thus, in the following section, we introduce *conformal prediction*, a framework that uses uncertainty estimation to make model predictions more reliable. Specific applications of conformal prediction are presented in Chapters 7 and 9.

4.3.2.1 Conformal Prediction

Conformal prediction (CP) is a method for reliability estimation that can be applied to both classification and regression models as long as they provide some notion of prediction (un-)certainty (see following section). While a conventional ML model yields point predictions, a model extended with CP yields sets of classes for classification or value intervals for regression instead. Intuitively, the size of the predicted set/interval for a given sample indicates how certain the model is in its prediction, with small sets and narrow intervals corresponding to increased confidence. The aim of CP is to generate predictions such that the true response of a sample is contained in the predicted set/interval with a probability of $(1 - \alpha)$, where $\alpha \in [0, 1]$ is the desired maximal error rate. This is also known as the CP *certainty guarantee*.
In the following, we present CP in detail. We first discuss which requirements a model and the used data need to fulfill in order to apply CP and to achieve the CP certainty guarantee. Afterward, we present the mathematical details of the CP workflow for classification and regression.

Requirements and Guarantees

CP can be applied to any classification or regression method, given that the method provides a notion of (un-)certainty for its predictions. For classification, many models are able to predict class probabilities that indicate prediction (un-)certainty. For random forests, e.g., the fraction of trees that voted for a certain class can be interpreted as a class probability. For regression, the idea of prediction (un-)certainty is less straightforward but later in this section we will show how it can be measured using quantiles.
When developing ML models, two disjoint datasets $\mathbf{Z}_{\mathbf{train}} = (\mathbf{X}_{\mathbf{train}}, \mathbf{y}_{\mathbf{train}})$ and $\mathbf{Z}_{\mathbf{test}} = (\mathbf{X}_{\mathbf{test}}, \mathbf{y}_{\mathbf{test}})$ are required for model training and testing, respectively. For CP, an additional dataset $\mathbf{Z}_{\mathbf{cal}} = (\mathbf{X}_{\mathbf{cal}}, \mathbf{y}_{\mathbf{cal}})$ is needed, called *calibration data* that is disjoint from both the training and test data. The calibration data is used to quantify the model uncertainty when making predictions for previously unseen samples. Conventionally, the

training and test data are required to be i.i.d. (cf. Section 4.1.1). For CP, this requirement is slightly relaxed to demand for $\mathbf{Z_{train}}$, $\mathbf{Z_{test}}$, and $\mathbf{Z_{cal}}$ to be *exchangeable* [267], meaning that the underlying joint probability distribution is invariant to finite permutation [268]. Lastly, the user needs to define the desired error rate $\alpha \in [0, 1]$. Increasing α generally decreases the size of predicted intervals/sets but risks an increase in erroneous predictions where the true response is not contained in the predicted set/interval.
Given exchangeable train, test, and calibration datasets as well as an error rate α, CP predictions are guaranteed to contain the true response with a probability of almost exactly $(1-\alpha)$. More specifically, for a sample $x_i \in \mathbf{X_{test}}$ where CP generated a prediction set/interval $\mathcal{C}(x_i)$, it is guaranteed that

$$1 - \alpha \leq Pr(y_i \in \mathcal{C}(x_i)) \leq 1 - \alpha + \frac{1}{N_{cal} + 1} \quad , \tag{4.74}$$

where y_i denotes the actual discrete/continuous response of x_i and N_{cal} denotes the number of samples in the calibration data. The term $Pr(y_i \in \mathcal{C}(x_i))$ is often referred to as the *model certainty*. For $N_{cal} \to \infty$, this certainty becomes exactly $(1 - \alpha)$.
Equation 4.74 is also known as the *marginal coverage property* of CP. It states that the coverage of at least $(1 - \alpha)$ holds on average for unseen data points. In general, however, we would like to guarantee the desired coverage not just on average but for any particular sample x_i, which is also known as *conditional coverage* [267]:

$$Pr(y_i \in \mathcal{C}(x_i)|x_i) \geq 1 - \alpha \tag{4.75}$$

Conditional coverage is a much stronger demand than marginal coverage. According to Vovk et al., it cannot always be achieved but can be approximated [269] (see also the Mondrian score discussed below).

In the following section, we describe the steps to perform CP for both classification and regression in detail.

Conformal Prediction Pipeline

The CP pipeline always consists of the same three steps, independent of whether classification or regression is performed. Below, we summarize these steps and describe their purpose intuitively, while mathematical details are described afterward.

1. Training: Train an ML model using the training samples in $\mathbf{Z_{train}}$.
2. Calibration:
   - Use the trained model to make predictions for all calibration samples in $\mathbf{Z_{cal}}$.

- Apply a scoring function to the prediction of each calibration sample to obtain a score distribution. Intuitively, the scoring function quantifies the prediction (un-)certainty of the model for previously unseen samples.
- Determine $\hat{q}$ as the $\frac{\lceil (N_{cal}+1)(1-\alpha) \rceil}{N_{cal}}$ quantile of the score distribution. For simplicity, $\hat{q}$ is often referred to as the *modified* $(1-\alpha)$ *quantile.*

3. Predictions:

    - Use the trained model to make predictions for all test samples in $\mathbf{Z_{test}}$.
    - Combine $\hat{q}$ with the model predictions to generate prediction sets (classification) or intervals (regression) that have the desired coverage. Intuitively, for classification, the prediction (un-)certainty for each test sample is compared to $\hat{q}$ to decide which classes should be contained in the predicted interval. For regression, $\hat{q}$ is employed to determine the size of the predicted interval for each test sample.

In the following, we present several scoring functions that can be applied for either classification or regression to derive $\hat{q}$ in the calibration step. For each score, we also detail how sets/intervals are generated using $\hat{q}$ in the prediction step. In Chapter 7, the application of these scores for drug sensitivity prediction is investigated and differences between the results obtained from different scoring functions are discussed.

Classification Scores

We first discuss three scoring functions that can be used to quantify prediction uncertainty for classification models with K classes $c_1, ..., c_K$. To measure uncertainty for individual samples, these scores rely on class probabilities. How these probabilities can be obtained from different ML models is discussed in Section 4.2.

True Class score: The True Class score was developed by Angelopoulos and Bates [267]. For a sample $x_j \in \mathbf{X_{cal}}$ with known class y_j, it denotes the probability of misclassification and is defined as:

$$s_{TC}(x_j, y_j) = 1 - \hat{Pr}(y_j|x_j) \tag{4.76}$$

Here, $\hat{Pr}(y_j|x_j)$ denotes the predicted probability of sample j belonging to class y_j derived from the model. Since y_j is the actual class of sample, the score becomes small if the model correctly assigns a high probability to y_j.

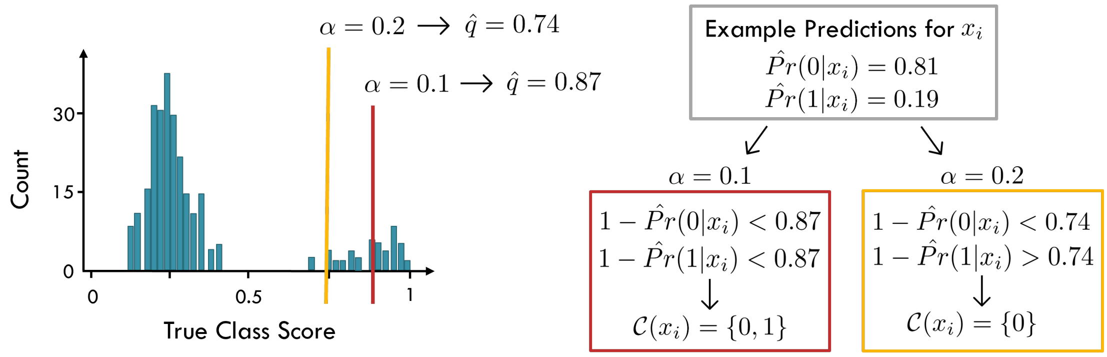


FIGURE 4.4: Impact of α on predicted sets. Consider a binary classification with classes 0 and 1. The left side of this figure depicts the distribution of the True Class score for the calibration data. For $a = 0.1$ and $\alpha = 0.2$, two different thresholds $\hat{q}$ can be derived from this distribution. On the right, an exemplary prediction for a sample x_i is shown, where changing α results in different sets $\mathcal{C}(x_i)$.

By applying s_{TC} to all calibration samples in $\mathbf{X_{cal}}$, we can generate a score distribution. After computing $\hat{q}$ as the modified $(1-\alpha)$ quantile of this score distribution, a prediction set $\mathcal{C}(x_i)$ for a previously unseen sample x_i is generated by including all classes c_k with $k \in \{1, ..., K\}$, for which $1 - \hat{Pr}(c_k|x_i)$ does not exceed $\hat{q}$:

$$\mathcal{C}(x_i) = \{c_k | 1 - \hat{Pr}(c_k|x_i) \leq \hat{q}, \forall k \in \{1, ..., K\}\} \tag{4.77}$$

Intuitively, we only include those classes in $\mathcal{C}(x_i)$ for which the probability of misclassification is sufficiently small according to $\hat{q}$. Figure 4.4 gives an example of how different choices of α influence the predicted sets for a binary classification with classes 0 and 1.

Summation score: The methodology behind the Summation score was first described by Angelopoulos and Bates [270] under the name *classification with adaptive prediction sets* based on ideas from [271] and [270]. To compute this score for a sample $x_j \in \mathbf{X_{cal}}$, all available classes $c_k \in \{c_1, ..., c_K\}$ are first sorted by their predicted class probabilities $\hat{Pr}(c_k|x_j)$ from highest to lowest. W.l.o.g, this sorted list for sample x_j is given by $u_j = [c_1, ..., c_K]$. The Summation score is then computed by going through this list and adding up the probabilities of all classes until the actual class y_j is reached:

$$s_{Sum}(x_j, y_j) = \sum_{o=c_1}^{y_j} \hat{Pr}(o|x_j) \tag{4.78}$$

This score is large if either x_j was correctly predicted as class y_j with a high probability or if x_j was misclassified. In the latter case, class y_j is located near the end of the list u_j and several class probabilities needed to be summed up before reaching y_j.
By applying s_{Sum} to all calibration samples in $\mathbf{X_{cal}}$, we can again generate a score distribution and derive $\hat{q}$ as the modified $(1 - \alpha)$ quantile. The prediction set $\mathcal{C}(x_i)$ for

a new sample x_i is then derived by first generating a sorted class list u_i as described above. Next, we successively add classes from u_i to $\mathcal{C}(x_i)$ in order of decreasing class probability until the sum of their probabilities exceeds $\hat{q}$:

$$\mathcal{C}(x_i) = \left\{ c_k \,\middle|\, k \in \{1, ..., \max_{k'} \left\{ \sum_{u=1}^{k'} \hat{Pr}(c_u|x_i) < \hat{q} \right\} + 1 \right\} \right\} \tag{4.79}$$

Mondrian score: The Mondrian score [272] aims to extend the marginal coverage guarantee of CP (cf. Equation 4.74) to hold not just on average (i.e., across all classes) but for each individual class $c_k \in \{c_1, ..., c_K\}$. Thus, even for classes c_k that may be underrepresented in the data, we want to guarantee that the predicted intervals for samples belonging to c_k contain this class with the desired probability. Mathematically, the prediction set $\mathcal{C}(x_i)$ for a new sample x_i should fulfill [267]:

$$Pr(y_i \in \mathcal{C}(x_i) \,|\, y_i = c_k) \geq 1 - \alpha \quad \forall k \in \{1, ..., K\} \tag{4.80}$$

Intuitively, the Mondrian score moves toward achieving conditional coverage (cf. Equation 4.75) by guaranteeing the desired coverage $(1 - \alpha)$ for each predefined class c_k in the data. However, true conditional coverage would require this coverage to hold for any subset of samples in the data, not just those subsets defined through the response classes.

As remarked above, extending the coverage guarantee to hold for each class can be especially useful when classes are highly imbalanced. In such cases, the desired coverage is typically only achieved for the most prevalent class(es) but not for the underrepresented one(s) [267]. This phenomenon can be observed in our analyses in Chapter 7

The idea behind computing the Mondrian score is to perform the calibration step for each class separately. This results in one $\hat{q}_k$ for each class c_k calculated on the subset of calibration samples belonging to class c_k. As a scoring function, for example, the True Class score, i.e., $1 - \hat{Pr}(y_j|x_j)$, can be employed (cf. Equation 4.76). The prediction set $\mathcal{C}(x_i)$ for a new sample x_i is then computed as follows:

$$\mathcal{C}(x_i) = \{c_k \,|\, 1 - \hat{Pr}(c_k|x_i) \leq \hat{q}_k, \forall k \in \{1, ..., K\}\} \tag{4.81}$$

A noteworthy property of Equation 4.81 is that $\mathcal{C}(x_i)$ is not required to contain the class with the highest prediction probability, which would be equal to the predicted class $\hat{y}_i$ in a conventional classifier. Even is a class c_k is predicted as the most probable, the corresponding prediction uncertainty may still be greater than the class-specific threshold $\hat{q}_k$, leading to the exclusion of this class from $\mathcal{C}(x_i)$.

Regression score

Romano et al. developed a regression score for CP that is based on quantile regression [273]. Even without CP, quantile regression can be used to estimate prediction certainty through generating prediction intervals: First, the training data is used to build a model $\hat{f}_{\frac{\alpha}{2}}$ to predict the $\frac{\alpha}{2}$ quantile of the response, and a second model $\hat{f}_{1-\frac{\alpha}{2}}$ to predict the $(1-\frac{\alpha}{2})$ quantile. Now, we may expect the interval $[\hat{f}_{\frac{\alpha}{2}}(x_j), \hat{f}_{1-\frac{\alpha}{2}}(x_i)]$ to contain the true response of a new sample x_i with a certainty of $(1-\alpha)$. However, this certainty is only based on the training data and, consequently, may not hold for previously unseen samples. Hence, Romano et al. developed a scoring function that allows adjusting the intervals generated by quantile regression to fulfill the desired coverage for previously unseen samples [273]:

$$s_{Qu}(x_j, y_j) = max \begin{cases} \hat{f}_{\frac{\alpha}{2}}(x_j) - y_j \\ y_j - \hat{f}_{1-\frac{\alpha}{2}}(x_j) \end{cases} \tag{4.82}$$

Here, $x_j \in \mathbf{X_{cal}}$ is a calibration sample with a known response y_j. The scoring function measures the distance between y_j and the nearest boundary of the predicted quantile regression interval for x_j. If y_j is included in the predicted interval, the sign of the score is negative, and otherwise, it is positive.

Again, we compute a score distribution of s_{Qu} using the calibration samples and calculate $\hat{q}$ as the modified $(1-\alpha)$-quantile of this distribution. The CP prediction interval for a new sample x_i is then given by:

$$\mathcal{C}(x_i) = [\hat{f}_{\frac{\alpha}{2}}(x_i) - \hat{q}, \hat{f}_{1-\frac{\alpha}{2}}(x_i) + \hat{q}] \tag{4.83}$$

If the calibration samples were contained in the quantile regression intervals with a probability below the desired coverage of $(1-\alpha)$, then $\hat{q} > 0$, and the intervals need to be widened to achieve the desired coverage. In contrast, if a coverage of exactly $(1-\alpha)$ was achieved by the quantile regression intervals, $\hat{q}$ is close to 0 and, thus, the intervals are unaltered. Finally, if $\hat{q} < 0$, the quantile regression intervals can be narrowed, making predictions more precise.

4.3.3 Interpretability

Especially in sensitive fields such as personalized medicine, where incorrect treatment choices can have severe consequences, being able to understand or explain the decision-making process of a model is crucial. The extent to which humans can understand a model and its decisions is commonly referred to as *interpretability* [30–32]. At first glance, this description of interpretability seems rather intuitive. Moreover, many publications on drug sensitivity prediction state they have achieved or at least attempted the development of an interpretable model [26]. However, it remains difficult to pinpoint exactly what properties make a model and its predictions interpretable or what specific strategies can be employed to interpret them [274]. Consequently, we developed a taxonomy of interpretability that summarizes common notions of the term, which is depicted in Figure 4.5. The aim was to derive a system that defines and distinguishes different types of interpretability, thus providing a designated vocabulary for describing interpretability and pointing out strategies for how models and predictions can be better understood.

Our taxonomy distinguishes two main areas of interpretability: (1) *transparency*, which denotes the inherent understandability of a model, and (2) *explainability*, which involves generating post-hoc explanations to understand a model's predictions. Each area comprises several subtypes of interpretability, which will be described in the following. Notably, a model can be both transparent and explainable at the same time. Note also that the different interpretability types are not necessarily clearly defined mathematical concepts and they are not always clearly separable from each other. Rather, they are supposed to provide different viewpoints on how to assess or enhance model interpretability.

In accordance with Lipton [275], we subdivide model transparency into three categories:

- **Simulatabilty** refers to the simplicity of a model as a whole. Two subtypes of simulatability can be distinguished: (1) simplicity in terms of the model size, e.g., the number of parameters and (2) simplicity in terms of the ease of performing manual inference in a reasonable time. For example, a human can typically easily recalculate the prediction of an individual decision tree. In contrast, the manual inference of a neural network's prediction with thousands of nodes and edges is barely possible.

- **Decomposability** focuses on the comprehensibility of individual components of a model, such as the input representation or calculation rules, and their correspondence to real-world phenomena, e.g., genes or pathways. Ideally, all components of a model should be comprehensible [275, 276]. However, often only parts of a model are understandable. Additionally, if the number of components is large and they have a complex interplay, the model as a whole is not necessarily interpretable, even though its individual components are comprehensible.

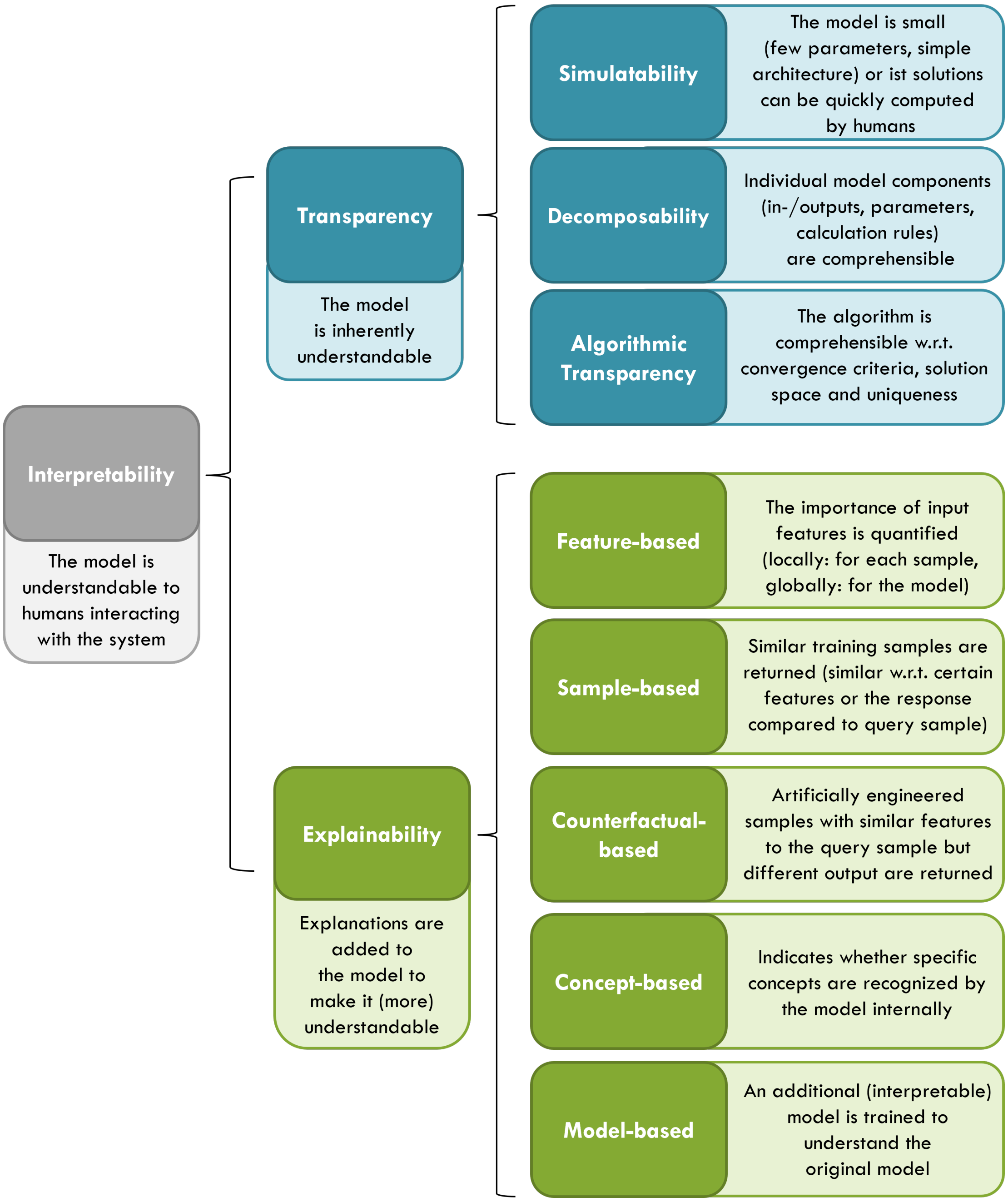

Interpretability
The model is understandable to humans interacting with the system
Transparency
The model is inherently understandable
Simulatability
The model is small (few parameters, simple architecture) or ist solutions can be quickly computed by humans
Decomposability
Individual model components (in-/outputs, parameters, calculation rules) are comprehensible
Algorithmic Transparency
The algorithm is comprehensible w.r.t. convergence criteria, solution space and uniqueness
Explainability
Explanations are added to the model to make it (more) understandable
Feature-based
The importance of input features is quantified (locally: for each sample, globally: for the model)
Sample-based
Similar training samples are returned (similar w.r.t. certain features or the response compared to query sample)
Counterfactual-based
Artificially engineered samples with similar features to the query sample but different output are returned
Concept-based
Indicates whether specific concepts are recognized by the model internally
Model-based
An additional (interpretable) model is trained to understand the original model

FIGURE 4.5: Taxonomy of interpretability. This figure depicts a taxonomy of interpretability in the context of ML, which we derived from the works of Lipton [275], Biran et al. [32], and Imrie, Davis, and van der Schaar [31]. We distinguish two types of interpretability, namely transparency (blue) and explainability (green), which can be further divided into three and five subtypes, respectively.

- **Algorithmic transparency** refers to the comprehensibility of the learning algorithm that is used in a model. This includes knowledge about the solution space and the shape of the error function or guarantees regarding the uniqueness of solutions. For linear models, for example, the shape of the error surface is known and it can be proven that model training converges to a unique solution, while this is not the case for complex deep neural networks [275].

For models to be transparent, they are often required to be simple. Consequently, they might be too simple to accurately model real-world tasks such as the prediction of drug responses based on high-dimensional cell line data. More sophisticated models that can model such complex tasks accurately enough are often of black-box nature, meaning they are not inherently transparent. For such models (but also for inherently transparent ones), post-hoc explanations can aid in interpreting a model's predictions and decision-making. The applicability of post-hoc explanation techniques is known as model *explainability*. Following the work by Imrie et al. [31], explainability can be subdivided into the following five categories of post-hoc explanations:

- **Feature-based explainability** measures the importance of input features, either locally for individual samples or globally for the entire model. Examples of global feature importance measures are the size of estimated coefficients for linear regression (cf. Section 4.2.1) or feature importance scores of random forests, which are, e.g., measured as the effect of randomly permuting an input variable on prediction correctness [236]. Examples of sample-specific feature importances include Shapley [277] and SHAP [278] values that measure how much each input feature contributes to an individual prediction. For two-dimensional data, *saliency maps* are commonly used for neural networks, e.g., to identify important pixels in an image [279].
- **Sample-based explainability** aims to identify training samples that the model treats similarly to a query sample of interest. An example would be training samples that reach the same leaf node in a decision tree or the nearest neighbors in a K-nearest neighbors model. After the identification of these samples, commonalities between the query sample and the similar samples can be investigated, e.g., to derive a common feature signature. Furthermore, if the samples that are similar to the query were often misclassified or have large prediction errors, this can be an indication that the prediction for the query sample may be unreliable.

- **Counterfactual-based explainability** involves the generation of artificially engineered samples that are similar to a query sample of interest but result in a different model output, i.e., a different classification or a strongly differing regression prediction. The aim is to determine which (potentially minor) alterations in the query's features affect the predicted response. For example, one could investigate the impact of manually in-/decreasing the expression of a drug's target gene in the model input.

- **Concept-based explainability** investigates whether certain concepts are recognized by a model. This type of explainability is arguably most difficult to grasp due to the arbitrary notion of *concepts* and their *recognition*. A concept can, e.g., be the presence of patterns in an image, such as stripes that aid the model in recognizing a zebra or irregular shapes in medical images that might indicate a tumor. Another example concept would be the presence of known biomarkers of drug response in the omics-features of a cell line.
  Recognition of a concept is typically investigated using the internal data representation of the model, e.g., the activation of certain nodes and edges in a neural network or the specific paths being taken in a decision tree. Typically, samples exhibiting a certain concept of interest are compared to samples not exhibiting the concept to investigate whether their internal representations differ. Automated methods for concept-based explainability such as TCAV [280] and CAR [281] currently focus mainly on neural networks.

- **Model-based explainability** aims to derive a second, more transparent (but generally less accurate) model from the original model, which should elucidate the decision process of the original model. An example would be a single decision tree that captures important decisions of an entire random forest.

These explainability types should not be seen as strictly separable concepts but rather as different viewpoints on how to approach explainability. In fact, most applications likely benefit from employing a combination of different explainability types. For example, after identifying training samples that are similar to a query sample of interest (sample-based explainability), one might want to identify common properties of these samples, which can be seen as a form of feature-based explainability. Similarly, the investigation of concept-based explainability may require the engineering of counterfactual samples that may share the presence of the investigated concept but still differ in their internal model representation and response.

Table 4.1: Realization of interpretability and reliability in drug sensitivity/synergy literature. This table compares 36 state-of-the-art approaches for drug sensitivity/synergy prediction regarding their use of different interpretability types (cf. Figure 4.5), as well as reliability estimation (white: regression, blue: regression and classification, green: classification, yellow: semi-supervised learning, orange: reinforcement learning).

	Transparency			**Explainability**					
Model	Simulatability	Decomposability	Algor. transparency	Feature	Sample	Counterfactual	Concept	Model	**Reliability**
Menden et al. (2013) [282]	✗	✗	✗	✗	✗	✗	✗	✗	✗
Zhang et al. (2015) [283]	✗	✓	✗	✗	✗	✗	✗	✗	✗
SRMF (2017) [284]	✗	✓	✗	✗	✗	✗	✗	✗	✗
HARF (2017) [285]	✗	✓	✗	✗	✗	✗	✗	✗	✗
Matlock et al. (2018) [286]	✗	✓	✗	✗	✗	✗	✗	✗	✗
TreeCombo (2018) [287]	✗	✓	✗	✓	✗	✗	✗	✗	✗
KRL (2018) [288]	✗	✗	✗	✗	✗	✗	✗	✗	✗
RWEN (2018) [38]	✗	✓	✗	✗	✗	✗	✗	✗	✗
CDRscan (2018) [289]	✗	✗	✗	✗	✗	✗	✗	✗	✗
QRF (2018) [290]	✗	✓	✗	✓	✗	✗	✗	✗	✓
NCFGER (2018) [291]	✗	✓	✗	✗	✗	✗	✗	✗	✗
DeepDR (2019) [292]	✗	✗	✗	✗	✗	✗	✗	✗	✗
netBITE (2019) [293]	✗	✓	✗	✗	✗	✗	✗	✗	✗
Deng et al. (2020) [294]	✗	✓	✗	✗	✗	✓	✗	✗	✗
Ahmed et al. (2020) [295]	✗	✓	✗	✗	✗	✗	✗	✗	✗
ADRML (2020) [296]	✗	✓	✗	✗	✗	✗	✗	✗	✗
PathDSP (2021) [297]	✗	✗	✗	✓	✗	✗	✗	✗	✗
REFINED CNN (2021) [298]	✗	✗	✗	✗	✗	✗	✗	✗	✗
GraphDRP (2021) [299]	✗	✓	✗	✓	✗	✗	✗	✗	✗
Precily (2022) [300]	✗	✗	✗	✗	✗	✗	✗	✗	✗
KBMTL (2014) [301]	✗	✗	✗	✗	✗	✗	✗	✗	✗
DeepSynergy (2018) [302]	✗	✗	✗	✗	✗	✗	✗	✗	✗
DeepCDR (2020) [303]	✗	✗	✗	✓	✗	✗	✗	✗	✗
Kim et al. (2021) [304]	✗	✗	✗	✗	✗	✗	✗	✗	✗
MatchMaker (2022) [305]	✗	✗	✗	✗	✗	✗	✗	✗	✗
SAURON-RF (2022) [39]	✗	✓	✗	✓	✗	✗	✗	✗	✗
reliable SAURON-RF (2023) [40]	✗	✓	✗	✗	✗	✗	✗	✗	✓
LOBICO (2016) [36]	✓	✓	✓	✓	✗	✗	✗	✗	✗
Stanfield et al. (2017) [177]	✗	✓	✗	✗	✗	✗	✗	✗	✗
SyDRa (2017) [306]	✗	✓	✗	✗	✗	✗	✗	✗	✗
HNMDRP (2018) [307]	✗	✓	✗	✗	✗	✗	✗	✗	✗
pLETORg (2018) [178]	✗	✗	✗	✗	✗	✗	✗	✗	✗
Deep-Resp-Forest (2019) [179]	✗	✗	✗	✗	✗	✗	✗	✗	✗
MERIDA (2021) [37]	✓	✓	✓	✗	✗	✗	✗	✗	✗
Dr.VAE (2019) [213]	✗	✗	✗	✗	✗	✗	✗	✗	✗
PPORank (2022) [215]	✗	✗	✗	✗	✗	✗	✗	✗	✗

In our review [26], where we first introduced our taxonomy, we also investigated the use of interpretability in 36 state-of-the-art approaches for drug sensitivity and drug synergy prediction. The results are shown in Table 4.1 and reveal that most interpretability types other than decomposability are heavily underexplored in the current drug sensitivity literature. Note that we already considered an approach *decomposable* if individual model components are understandable, which is a relatively weak requirement for interpretability.

4.3.4 Tailoring Models to the Task and Data at Hand

Another important aspect of trustworthiness is how well a model is tailored to the task it should perform. It is essential to recognize the assumptions and limitations of a model and its underlying data to contextualize its predictions. In the following, we use drug sensitivity prediction as an example to describe ways that models can be tailored to a certain task. Additionally, we highlight how the available training data can impose certain limitations on a model. These limitations should be considered when applying the model and interpreting its predictions.

4.3.4.1 Model Inputs

As discussed in Chapter 3, data for studying anti-cancer drug responses is often not obtained directly from humans but from model systems such as cancer cell lines. Thus, one should be aware of the fact that certain mechanisms, such as drug metabolization, cannot be fully captured by this data (cf. Section 3.1). For example, a treatment that is predicted as effective in a cell line due to the presence of a certain mutation may not be effective in a patient with the same mutation, if the patient's body metabolizes the drug before it can take effect. Consequently, the explanation provided by the model may not help in understanding the patient's drug response. Similarly, including drug properties that might be important for drug metabolization in humans might not be efficiently learned by a model trained on cell lines. Thus, model interpretability may be limited for such features.

The omics-measurements that are typically used to characterize cell lines for drug sensitivity prediction are very high-dimensional. In contrast, the number of available cell lines is relatively small. This leads to a phenomenon called the curse of dimensionality, where the number of samples required to sufficiently cover the feature space (e.g., to reliably estimate densities) increases exponentially with the number of features [308]. Consequently, a high-dimensional feature space may only be sparsely populated if the number of available samples is low. Training a model on such data can lead to overfitting and poor performance. Furthermore, the extremely large number of inputs can hamper model interpretation. Consequently, the number of considered features is commonly reduced using *dimension reduction* techniques.

Dimension Reduction: Two groups of dimension reduction algorithms can be distinguished: *feature selection* and *feature extraction* techniques [309]. Feature selection algorithms aim to choose a subset of interesting features from a given feature set. They can be partitioned into *filters*, *wrappers*, and *embedded methods* [309]. Filter methods can be seen as a pre-processing step, where features are chosen before training an ML model. Common examples include selecting the features with the highest variance, or strongest correlation to the response. Alternatively, features can be derived from literature or expert knowledge. In contrast, wrappers train ML models using different feature sets and then select the best features based on the employed model error. An example is *recursive feature elimination* [310], where an initial model is trained using all available features. Afterward, the feature with the lowest importance in the model is removed, and the model is retrained. This process is repeated until the desired number of features is reached. Lastly, embedded methods are ML algorithms where the feature selection is integrated into the model training. An example is the elastic net discussed in Section 4.12, where the coefficients of less informative features can be set to 0, thereby excluding the corresponding features from the model.

In contrast to feature selection, feature extraction aims to transform a high-dimensional dataset into a lower-dimensional representation, which still retains properties of the original data space. Common examples are autoencoders [311] or principal component analysis [312]. While feature extraction does not require discarding potentially valuable features, the resulting lower-dimensional data representation is often not easily interpretable, as each calculated feature is a combination of features from the original feature space.

In Chapter 5, we systematically compare the performance of drug sensitivity prediction models trained using features derived from nine different dimension reduction techniques, including six filter methods and three feature extraction approaches.

4.3.4.2 Model Outputs

As discussed in Chapter 3, drug response is commonly quantified using measures like the IC50 or AUC value. If the task at hand is solely to predict drug responses, these measures may be well-suited as model outputs. If, however, the task is to perform drug prioritization, i.e., to obtain a list of drugs sorted by effectiveness for a given sample, such measures cannot be used as they are not comparable across drugs (cf. Section

3.2.4.1). In Chapter 7, we introduce a novel drug response measure that is comparable across drugs and, thus, suited to perform drug prioritization.

4.3.4.3 Choice of ML Algorithm

Based on the available data, an ML algorithm should be chosen whose assumptions match the task at hand. For example, one might assume that the relationship between molecular characteristics of cell lines and their drug response is not linear and should, thus, not be modeled using linear approaches. However, our results in Chapter 5 show that elastic net models can frequently outperform non-linear models for drug sensitivity prediction. A factor that can heavily influence the performance of a model is the amount of available data: For complex models with many parameters, such as neural networks, the available data may be insufficient to effectively train these parameters (cf. *bias-variance trade-off* in Section 4.1.1 and *approximation uncertainty* in Section 4.3.2).

4.3.4.4 Performance Assessment

Once a model is trained, its performance can be assessed. For this step, it is again crucial to precisely formulate the task that a model should fulfill and to evaluate its performance accordingly. For example, if the task is to build a model that can accurately predict the drug response of previously unseen cell lines, the data used for evaluation should not contain any cell lines that are also included in the training data, as this might overestimate the model's performance [313]. For multi-drug models, e.g., it is common that the same cell lines (combined with different drugs) are used for both training and testing (cf. Chapter 8). For such models, performance can be evaluated separately for seen and unseen cell lines to gain insight into whether predictions for one group are more accurate than the other. In a similar vein, it is often beneficial to investigate model performance on various data subsets, e.g., cell lines from a specific tissue or drugs targeting the same pathway. This can reveal potential biases in the model that make predictions for certain cases more accurate and reliable than others. These biases should be kept in mind when applying the model. During model training, they can actively be counteracted, e.g., by using sample-specific weights (cf. Section 4.2).

In Chapter 6, we discuss one well-known bias of drug sensitivity prediction models in detail, namely the poor prediction of samples with a sensitive (as opposed to a resistant) drug response, which arises from an underrepresentation of these cases in drug screening data.

4.3.4.5 Model Application

A final goal in drug sensitivity research is to deploy the developed ML models into decision support systems for personalized treatment recommendation. Even if the underlying model is well suited for this task, a system that is difficult to use or fails to provide information in an intuitive manner will likely not be trusted by its users. Consequently, user-friendliness should be a main concern when developing a decision support system. Therefore, the requirements that different user groups, e.g., medical doctors, patients, or bioinformaticians place on the system and their ability to understand the provided information should be taken into account.

Chapter 5

A Benchmarking of Machine Learning and Dimension Reduction Methods

An essential requirement for any ML model is a good predictive performance. Consequently, their performance is a main selling point of many newly published approaches for drug sensitivity prediction. In recent years, the most widely used ML algorithm for drug sensitivity prediction has been neural networks, where increasingly complex models with carefully crafted architectures have claimed highly promising results, e.g., [215, 289, 294, 299, 314, 315]. However, Li et al. found that many complex deep learning approaches are, in fact, not superior in performance to simple feed-forward neural networks or random forests for drug sensitivity prediction [24]. And even though there are further indications that tree-based methods can outperform neural networks, especially in data-sparse settings [316–318], we are only aware of relatively few random forest-based approaches for drug sensitivity prediction [39, 179, 285, 290, 293]. Similarly, linear models such as elastic net have barely been investigated in this field [38], despite the fact that their straightforward interpretability would be a major benefit: Keeping models as comprehensible as possible is desirable for studying the relationship between cellular features and drug response and for creating interpretable predictions to enable ML-based clinical decision support. Consequently, the question arises of how complex our prediction models actually need to be, especially given the currently limited availability of training data (cf. Chapter 2). Additionally, we may ask how simple(r) models perform in comparison to complex ones.

As discussed in Chapter 4.3.1, the predictive capabilities of ML models are typically assessed through error measures like the mean squared error (MSE). While some form of performance comparison with existing approaches is often conducted for newly published approaches, the significance of such comparisons can be hampered by several factors, including (1) evaluating models without proper hyperparameter optimization, (2) data leakage, where information from test samples influences model training, often leading to artificially inflated performance measures [319], and (3) differences in the number of input features, as well as their type and representation.

Regarding point (3), we have already discussed in Section 4.3.4 that the high dimensionality of the cell line data makes the use of dimension reduction (DR) typically indispensable to counteract the curse of dimensionality. Even though DR methods are routinely applied, it is, unfortunately, rarely reported in publications whether different DR methods were considered or compared. This issue is illustrated in Table 5.1, where we summarize information on the used DR techniques in 32 state-of-the-art methods for sensitivity prediction. Overall, given the plethora of available ML algorithms and DR techniques, a systematic and fair performance evaluation is needed.

Several previously published benchmarkings already move in the direction of achieving this goal: Chen et al. benchmarked several state-of-the-art deep learning approaches for drug sensitivity prediction [25], but they did not investigate the impact of different DR methods. Jang et al. analyzed the performance of different input features and ML algorithms for drug sensitivity prediction [23]. However, they focused on inputs from different omics types rather than different DR techniques. Koras et al. compared several feature selection methods [320], but they did not account for size differences in the investigated feature sets, which might impact performance more significantly than the chosen feature selection method.

The analyses presented in this chapter aim to overcome the limitations of existing benchmarkings. Using gene expression values and drug response measures from the *Genomics of Drug Sensitivity in Cancer* (GDSC) database (cf. Section 3.4), we performed a systematic benchmarking comparing four ML algorithms in combination with nine DR techniques. We additionally varied the size of feature sets and performed extensive hyperparameter tuning, resulting in more than 16,000,000 trained models for 179 compounds. To the best of our knowledge, this is the largest drug sensitivity benchmarking to date in terms of the number of investigated DR methods, feature numbers, and hyperparameters.

In the following, we first describe the data we employed for our analyses and how the models were trained, including a detailed description of the used DR methods. We then compare the models regarding prediction accuracy, runtime, and interpretability. Our findings reveal that the choice of both the ML algorithm and DR method substantially impacts performance. We also discuss how performance trade-offs between models can be assessed. Furthermore, we show how model hyperparameters can be tuned to improve predictions for the most sensitive cell lines, a challenge that is extensively discussed and addressed in Chapter 6.

Table 5.1: Overview of the use of DR in different ML approaches for drug sensitivity prediction. For 32 publications, this table lists the DR technique that was applied to derive cell line-based input features. Similar techniques are grouped by color. Additionally, it is denoted whether a performance comparison of different DR methods was performed.

Publication	DR for cell line features	DR comparison
Menden et al. (2013) [282]	literature-based	✗
LOBICO (2016) [36]	literature-based	✗
Stanfield et al. (2017) [177]	literature-based	✗
CDRscan (2018) [289]	literature-based	✗
Deng et al. (2020) [294]	literature-based	✗
MERIDA (2021) [37]	literature-based	✗
PPORank (2022) [215]	literature-based	✗
GraphCDR (2021) [321]	literature-based, late integration embedding*	✗
Dr.VAE (2019) [213]	literature-based, autoencoder	✗
DeepDR (2019) [292]	autoencoder	PCA†
NeRD (2022) [322]	autoencoder, late integration embedding*	✗
GADRP (2023) [323]	autoencoder	PCA†
mVAEN (2023) [324]	autoencoder	PCA†
SAURON-RF (2022) [39]	min.-redundancy-max.-relevance	✗
reliable SAURON-RF (2023) [40]	min.-redundancy-max.-relevance	✗
Matlock et al. (2018) [286]	RELIEFF algorithm	✗
HARF (2017) [285]	RELIEFF algorithm	✗
QRF (2018) [290]	correlation, random forest feature importance	✗
RWEN (2018) [38]	elastic net	✗
NetBiTE (2019) [293]	feature biasing	✗
SRMF (2017) [284]	matrix factorization	✗
PathDSP (2021) [297]	pathway enrichment	✗
RAMP (2022) [315]	network embedding*	✗
MOLI (2019) [314]	variance, late integration embedding*	early integration*
Deep-Resp-Forest (2019) [179]	random	✗
Zhang et al. (2015) [283]	none	✗
NCFGER (2018) [325]	none	✗
HNMDRP (2018) [307]	none	✗
ADRML (2020) [296]	none	✗
KRL (2018) [288]	none	PCA†
Rahman and Pal (2016) [326]	unknown	✗
GraphDRP (2022) [299]	unknown	✗

[*] These approaches use neural networks to derive a lower-dimensional representation of multi-omics cell line features. We listed this as a type of DR, since it can be seen as an embedded feature extraction.

[†] PCA: principal component analysis

To account for the goal of interpretable models, we characterize the four investigated ML algorithms in terms of *transparency* [275] and *explainability* [31] as introduced in Chapter 4.3.3. Lastly, our analyses using a multi-omics multi-task deep learning approach by Chiu et al. [292] prove that even complex prediction models can benefit substantially from using different DR methods. However, we also show that such complex models can still be outperformed by standard ML models, even with small feature numbers.

Author Contributions

The content and text of this chapter are based on the following publication:

Eckhart, L., Lenhof, K., Rolli, L. M., & Lenhof, H. P. (2024). *A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.* Briefings in Bioinformatics, 25(4), bbae242. DOI: 10.1093/bib/bbae242

The majority of the research presented in this publication was conducted by me. I also drafted the publication manuscript. Kerstin Lenhof was involved in designing the study and implemented some of the data processing and dimension reduction methods. All authors analyzed the results and reviewed the manuscript.

5.1 Materials and Methods

In the following, we first describe the dataset we used to perform this benchmarking and discuss the investigated models and the training process. Afterward, we present the nine analyzed dimension reduction methods.

5.1.1 Dataset

Data for our analyses was obtained from the GDSC database (Release 8.3), which we described in detail in Section 3.4. More specifically, we downloaded normalized expression values for 17,419 genes and drug screening data in the form of logarithmized IC50 values for all 198 drugs screened in the GDSC2 dataset. Out of these 198 drugs, we only considered those 179 drugs for which sensitivity measures for at least 600 cell lines are provided (cf. Appendix Table C.1).

5.1.2 Model In- and Outputs

We trained drug-specific regression models that predict the logarithmized IC50 values of cell lines from their gene expression data using four ML algorithms: random forests, neural networks, boosting trees, and elastic nets. A detailed description of these algorithms can be found in Section 4.2. Model inputs were generated using six feature selection (FS) and three feature extraction (FE) techniques described below that select or compute input features based on normalized gene expression values. Some of the FS methods additionally consider the IC50 values to determine the most informative features. To investigate how the number of input features k affects the model performance, we generated input feature sets for each $k \in \{1, 2, 3, ..., 25, 50, 100, 200, 300, 400, 500\}$. In the following, we will refer to one *setting* as one combination of ML algorithm, DR technique, and number of inputs k used to train a certain model.

5.1.3 Model Training and Testing

For each drug, we divided the available cell lines into a training set (80% of cell lines) and a test set (20%). On the training set, we performed a 5-fold cross-validation (CV) to determine the best-performing hyperparameters of the ML model using the mean squared error (MSE, cf. Equation 4.65) as error measure. The investigated hyperparameters are provided in Table 5.2. For each hyperparameter combination, one final model is trained on the complete training data, and its performance is evaluated on the test set.

This procedure is performed separately for each *setting* (i.e., combination of ML algorithm, DR method, and number of inputs). As the training and test data (as well as the data in each CV fold) are identical across all settings for one drug, the performance of different settings can be compared directly. Note that input features are selected/computed using only samples in the training set (both for the CV and the training of the final model), such that the test data does not influence the choice of features.

5.1.4 Dimensionality Reduction Techniques

As discussed in Section 4.3.4, DR techniques can be separated into feature selection (FS) and feature extraction (FE) methods. FS algorithms aim to choose a subset of interesting features from a given feature set. Here, we only consider FS methods known as *filters* (as opposed to *wrappers* or *embedded methods*, cf. Section 4.3.4), where features are selected before applying any ML algorithm. In contrast to FS, FE aims to transform a high-dimensional dataset into a lower-dimensional representation, which still retains the

Table 5.2: Overview of all ML algorithms investigated for the training of models, including the used R/Python packages and tuned hyperparameters. The last column denotes the number of tested hyperparameter combinations. Unless stated otherwise, we employed the default parameters of each algorithm in their respective package. Further information on the architecture and hyperparameters of the trained neural networks can be found in Appendix Table C.2.

Model	Parameter	Value(s)	Combinations
Elastic net (glmnet 4.1.3 [327])	alpha	$[0, 1]$ in steps of 0.1	$11 \cdot 20 = 220$
	lambda	10^v, v being 20 equally spaced values $\in [-2, 2]$	
Random forest (ranger 0.13.1 [328])	mtry	$[1, 25]$ in steps of 2, $[40, 200]$ in steps of 20	up to 22
Boosting trees (gbm 2.1.8 [329])	n.trees	1-20	$20 \cdot 5 = 100$
	interaction.depth	1-5	
Neural network (tensorflow 1.13.1, keras 2.3.1 [330, 331])	hidden layers	1, 2, 3	$3 \cdot 2 \cdot 2 = 12$
	activation function	tanh, ELU (none in last layer)	
	dropout	10%, 30%	

properties of the original data space. While FE does not require discarding potentially valuable features, the resulting low-dimensional data representation is often not easily interpretable, as each calculated feature may contain information of several combined features from the original feature space.

5.1.4.1 Feature Selection Techniques

Randomized Feature Selection: We generated randomized feature sets by randomly sampling gene sets of size k from all genes with expression values provided in the GDSC. To get a more stable estimate of the prediction errors for random features, we generated ten random feature sets for each k and averaged the error measures of the ten corresponding models.

Literature-based Feature Selection: For the literature-based FS, we retrieved a list of cancer driver genes from the IntOGen website (Release 2020-02-01) [332]. We only considered genes for which expression values are provided in the GDSC. Additionally, genes with warnings in the IntOGen database (e.g., genes that are known artifacts) were removed. Next, we sorted the remaining 476 genes according to their smallest IntOGen tier from tier 1 (i.e., genes with the strongest evidence of being cancer drivers) to tier 3. Within each tier, genes were sorted descendingly according to the number of cohorts (i.e., datasets) for which they have been reported as a cancer driver. From the beginning of this sorted gene list, the first k genes were chosen.

Variance-based Feature Selection: For this FS, we chose the k genes, for which the variance of expression values was the largest.

Correlation-based Feature Selection: For this FS, we chose the k genes with the highest absolute Pearson correlation coefficient (cf. Equation 4.66) between the expression values of each gene and the IC50 values of the corresponding cell lines.

Enrichment-based Feature Selection: We developed this FS to identify genes whose up-/downregulation is linked to sensitivity/resistance to a given drug. First, we determine for each gene whether it is up- or downregulated in each cell line. To this end, we calculated gene-specific z-scores [333]: Let x_{ig} denote the expression of gene g in cell line $i \in \{1, ..., N\}$. Then, the corresponding z-score z_{ig} is defined as:

$$z_{ig} = \frac{x_{ig} - \bar{x}_g}{s_g} \tag{5.1}$$

$$\text{with} \;\; \bar{x}_g = \frac{1}{N} \sum_{i=1}^{N} x_{ig} \tag{5.2}$$

$$\text{and} \;\; s_g = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ig} - \bar{x}_g)^2} \tag{5.3}$$

We consider a gene upregulated (downregulated) in a cell line if its z-score is larger (smaller) than the 95% (5%) percentile of a standard normal distribution.

To identify genes where deregulation is linked to drug sensitivity or resistance, our approach uses *gene set enrichment analysis* (GSEA), particularly the unweighted GSEA algorithm by Subramanian et al. [334]. GSEA tests for the accumulation of a certain feature (here: up-/downregulation of a gene) at either end of an ordered list of samples (here: cell lines): Cell lines are sorted in ascending order based on their IC50 for the drug of interest to obtain a ranked list. Next, two Kolmogorov-Smirnov tests [335, 336] are conducted for each gene to determine whether cell lines in which the gene is (1) up- or (2) downregulated are enriched at either end of the list, respectively. Each test yields a p-value and a direction denoting whether the enrichment occurred at the top or bottom of the list. We adjusted the p-values using the Benjamini-Hochberg procedure

[337] separately for all up- and all downregulated genes, respectively.

This procedure results in four lists of genes: genes that are up-/downregulated among the most sensitive/resistant cell lines, respectively. To obtain the k most important features, we proceeded as follows: For each list, we order genes from smallest to largest p-value and assign a rank to each gene, starting at 1. If a gene occurs in multiple lists, we keep it only in the list with the smallest rank. Next, we merge all lists by sorting genes according to their rank, and p-values are used to break ties. From the beginning of this merged list, we then select the first k features.

MRMR Feature Selection: This FS method is based on the *minimum-redundancy-maximum-relevance* (MRMR) principle, which aims to select features with a strong dependence on the response variable (i.e., large *relevancy*) but weak dependence on each other (i.e., small *redundancy*) [338]. Our implementation is based on a greedy heuristic by Kwak and Choi [339] that iteratively selects features, starting with the most informative ones. As dependence-measure, the mutual information I is employed. For two discrete random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, the mutual information $I(X;Y)$ measures how much the entropy of X, i.e., the average uncertainty associated with the variable's potential states, decreases when we have knowledge of Y [340]. Mathematically, the entropy of X, $H(X)$, is defined as [340]:

$$H(X) = -\sum_{x \in \mathcal{X}} Pr(x) \cdot log(Pr(x)) \tag{5.4}$$

Here, $Pr(x)$ is the probability of X taking value x. Next, the conditional entropy, $H(X|Y)$, measures the average amount of uncertainty that remains in X when we have knowledge of Y [340]:

$$H(X|Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} Pr(x,y) \cdot log\left(\frac{Pr(x,y)}{Pr(y)}\right) \tag{5.5}$$

Here, $Pr(x,y)$ is the joint probability of X taking value x and Y simultaneously taking value y. Based on these definitions, the mutual information $I(X;Y)$ is computed as:

$$I(X;Y) = H(X) - H(X|Y) \tag{5.6}$$

For independent variables, $I(X;Y) = 0$, and larger values of I indicate an increasing dependence.

To perform the MRMR FS, let F denote the set of all potential input features (i.e., genes) and Y the response variable (i.e., ln(IC50) values for a given drug). Here, both gene expression and ln(IC50) values were discretized using an equal-width binning with six bins. Furthermore, let S be the list of selected features, which is initially empty. In each iteration of the algorithm, the feature $X_i \in F$ that maximizes the following term is added to S and removed from F:

$$\max_{X_i \in F} \left\{ I(Y;X_i) - \sum_{X_j \in S} \frac{I(Y;X_j)}{H(X_j)} \cdot I(X_i;X_j) \right\} \tag{5.7}$$

Intuitively, $I(Y;X_i)$ measures the relevance of feature X_i, i.e., how much we can learn about our response Y by knowing that feature. The remaining part of the term measures the redundancy: For each already selected feature $X_j \in S$, we measure its dependence with X_i, i.e., $I(X_i;X_j)$, which is additionally weighted by a factor. If the mutual information between X_i and X_j is large, the features carry similar information. Thus, if feature X_j is already selected, also selecting X_i would not provide much additional information but rather add redundancy to S.

This procedure is repeated until k features are selected, i.e., until $|S| = k$. The result is a list of k features ordered by importance. To keep the runtime manageable, we limited F to the 1000 genes with the largest mutual information to the response.

As the presented approach is a greedy heuristic, the selected features are not guaranteed to provide an optimal solution to the MRMR problem for a given k. Hence, we additionally implemented an MRMR-based FS as a quadratic optimization program (QP). However, the high runtime of this approach only allowed us to compute feature sets for $k \leq 5$ in a reasonable time ($<$ 500 seconds for a single k on a single training dataset). Additionally, we found no improvement in test MSE when using the features selected by the QP instead of the heuristic as inputs for our ML models. Consequently, the QP is not discussed further here but can be found in Appendix C.1.

5.1.4.2 Feature Extraction Techniques

Principal Component Analysis: Principal component analysis (PCA) is a method that projects the original data into a lower-dimensional feature space through linear transformation [312]: Each transformed feature – also called a *principal component* (PC) – is a linear combination of the original features. The first PC is defined as that linear feature combination that explains most of the variance observed in the data. Each subsequent PC is then constructed to explain most of the remaining variance that was not explained by the previous component(s). Additionally, all PCs are designed to be orthogonal to each other.

We performed a PCA using the expression values of all cell lines in the respective training set to extract a lower-dimensional representation of these cell lines using the first k PCs. We then used the transformation calculated on the training data to project the test set cell lines into the same k-dimensional space.

PASL: PASL (Pathway Activity Score Learning) by Karagiannaki et al. [341] is a DR approach that aims to produce (biologically) interpretable features. Given a feature matrix (here: gene expression data) and predefined feature sets (here: genes belonging to a certain pathway), PASL projects the data into a latent space, where each newly constructed feature is a linear combination of features from one of the predefined feature sets. This approach is similar to PCA, with the constraint that each linear combination can only use features (genes) from one of the feature subsets (pathways). The computed features can then be interpreted as pathway activity scores for each sample. Just like PCA, PASL computes features in an ordered manner, such that the features explaining most of the variance in the data are computed first. We applied PASL with default parameters to the training data to generate k pathway features. As feature sets, we considered the same data as Karagiannaki et al., namely pathways from KEGG [342], Reactome [343], and BioCarta [344]. Analogously to the PCA, we applied the linear combinations computed by PASL to the test cell lines to obtain their representation in the new feature space.

Autoencoder: An autoencoder is a type of neural network that can encode data into a lower dimension [311]. It consists of an *encoding* part, which generates the lower-dimensional representation of the input, and a *decoding* part, trained to reconstruct the original inputs from the encoded representation. The aim is to find an encoding that preserves the core information of the original features sufficiently well so that it can be reconstructed using the decoder. After training the entire network, only the encoder is used to generate the lower-dimensional (here: k-dimensional) representation of the inputs for training and test data.
As one autoencoder has to be trained for each drug, training dataset, and k separately, the high runtime (on average 4.5 minutes for a single model) did not permit us to perform any hyperparameter tuning. Note that tuning would require an additional CV nested inside the main CV described in Section 5.1.3. The used hyperparameters and network architecture are described in Appendix Table C.3.

5.1.5 Implementation

Elastic nets, random forests, and boosting trees were trained in R, while neural networks (including the autoencoders) were trained in Python. The respective packages and hyperparameters are listed in Table 5.2, and Appendix Tables C.2 and C.3 provide further information. Most DR approaches were also implemented in R, namely the random, literature-based, variance-based, and correlation-based FS, as well as the PCA. For the latter three, the *stats* package v.3.6.3 [345] was used. For PASL, we employed the MATLAB code provided by Karagiannaki et al. [341] and the *matlabr* R package v.1.5.2 [346]. To perform the enrichment-based FS, we used the C++ framework [347] of the GeneTrail tool suite [348] to compute the enrichments, which were then further processed in R. Finally, the MRMR-based FS [339] was implemented in C++. Our implementations are available at `https://github.com/unisb-bioinf/ML_DR_Benchmarking_Drug_Sensitivity_Prediction`.

5.2 Results

We trained drug-specific models that predict logarithmized IC50 values for 179 drugs of the GDSC2 dataset using four ML algorithms (random forests, neural networks, boosting trees, elastic nets) and the nine DR techniques presented above. To investigate how the number of input features k impacts performance, we trained models for each $k \in \{1, 2, 3, ..., 25, 50, 100, 200, 300, 400, 500\}$. The hyperparameters of each algorithm (cf. Table 5.2) were tuned using a 5-fold CV on the training data.
Investigating each combination of ML algorithm, DR technique, and k results in a total of 1,116 models per drug (after hyperparameter tuning), with the following exceptions:

- We only trained neural networks for the 50 drugs with the most available cell lines due to their high runtime (cf. Figure 5.6 and Appendix Table C.1).
- For elastic net, no models for $k = 1$ exist since the used R package *glmnet* [327] only allows feature sets with $k \geq 2$.
- For the IntOGen features, results for $k = 500$ do not exist since our filtered IntOGen list (cf. Section 5.1.4) consists of 476 features only.
- For the PCA features, results for $k = 500$ do not exist since most CV training sets contain less than 500 samples, and the number of principal components computed by the *stats* R package [345] is limited by the number of input samples.

In the following, we first analyze the statistical performance of the investigated ML algorithms and DR techniques. Then, we assess the trade-offs between models of different complexity. Next, we show how the prediction of sensitive cell lines can be improved by using different error measures for hyperparameter tuning. We then compare the four ML algorithms regarding their runtime and interpretability and exemplarily investigate features derived from the MRMR FS and their importance in elastic nets. Lastly, we show that even complex prediction models benefit from DR by investigating a deep learning method by Chiu et al. [292].

5.2.1 Average Test MSE

We first analyzed which ML method(s) and DR approach(es) yield the smallest test error across all investigated drugs. As discussed in the previous section, we trained one model for each combination of drug, ML algorithm, DR technique, and feature number k. In Figure 5.1A, we compare the performance of the different ML algorithms: For each k, the figure shows the mean test MSE of all models that use the same ML algorithm, i.e., we average the performance of each algorithm across drugs and DR methods. Overall, errors decrease as k increases. The decrease is most drastic from $k = 1$ to $k = 10$ and again when $k \geq 50$, especially for elastic net. For $k > 8$, elastic net and random forest yield the smallest MSE, while neural networks consistently perform the worst. A comparison of all algorithms using only the 50 drugs we used to train the neural networks is provided in Appendix Figure C.3 and shows the same trends.

In Figure 5.1B, we analogously visualize the mean test MSE of all models that use the same DR technique. The MRMR and correlation-based FSs yield the smallest test MSE, followed by PCA. Interestingly, for $k > 200$, the average error of PCA-based models increases. More detailed results for each combination of ML algorithm and DR method are provided in Appendix Figures C.1 and C.2. There, it can be seen that using PCA with $k > 200$ only increases errors for boosting trees and random forests but outperforms all other DR methods for elastic net and neural networks.

5.2.2 Best-Performing Settings

In this section, we analyze which combinations of ML algorithm and DR technique resulted in the smallest test MSE for each drug and how often a certain combination performed best. To this end, we identified the best-performing ML and DR combination for each drug and feature number k. Figure 5.2 visualizes, for each k, the number of drugs for which a certain ML algorithm and DR technique performed best. Figure 5.2A shows that random forest and elastic net are the ML algorithms that performed best most frequently, with elastic net being more dominant for large feature numbers. Larger feature sets likely contain more redundant or uninformative features, which might be easily ignored by elastic net models but not by random forests since a random subset of features is selected in each node (cf. Section 4.2.4.2).

Regarding the DR techniques (cf. Figure 5.2B), PCA and MRMR are the most successful, followed by the correlation-based FS. For large k, PCA is the best-performing approach. Figure 5.2C shows how often each specific combination of ML and DR was the best-performing. In line with the previous results, elastic net and PCA are by far the most successful combination.

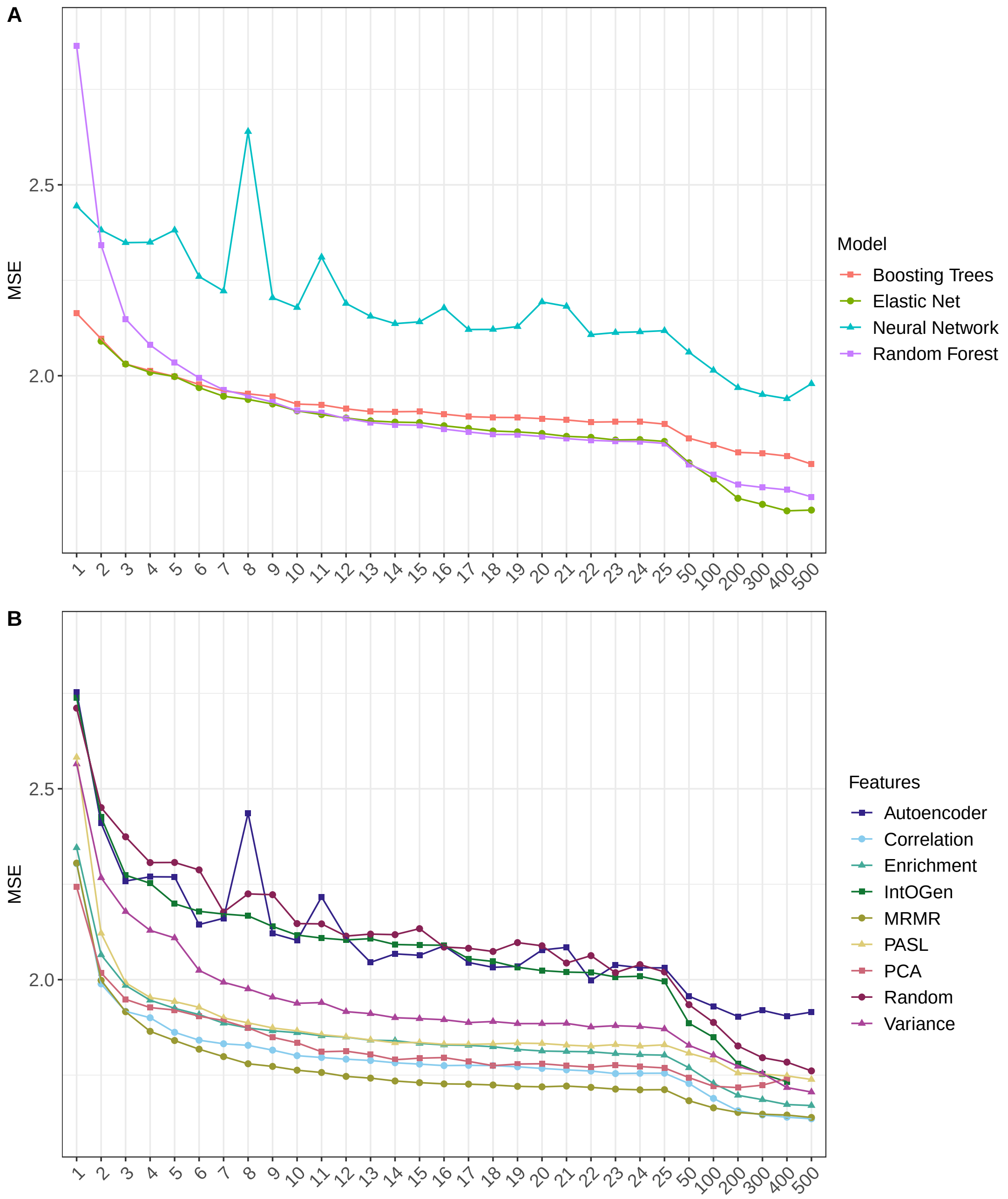


Figure 5.1: Average test MSEs. We consider one model for each combination of the 179 investigated drugs, four ML algorithms, nine DR techniques, and 31 feature numbers k (with the exceptions listed at the beginning of Section 5.2). Subfigure A depicts the average test MSE of these models for each ML algorithm and k, while Subfigure B depicts the average test MSE for each DR technique and k. The x-axis denotes the number of input features, the y-axis denotes the mean test MSE, and the coloring represents the different ML algorithms or DR techniques.

Figure 5.3 shows the same results as Figure 5.2 but limited to only the k that resulted in the best performance for each drug. For almost all drugs, large feature numbers ≥ 100 yielded the best results. The combination of elastic net and PCA was the best-performing for 76 of the 179 investigated drugs. However, all DR techniques other than the random FS had the best performance for at least some drugs.

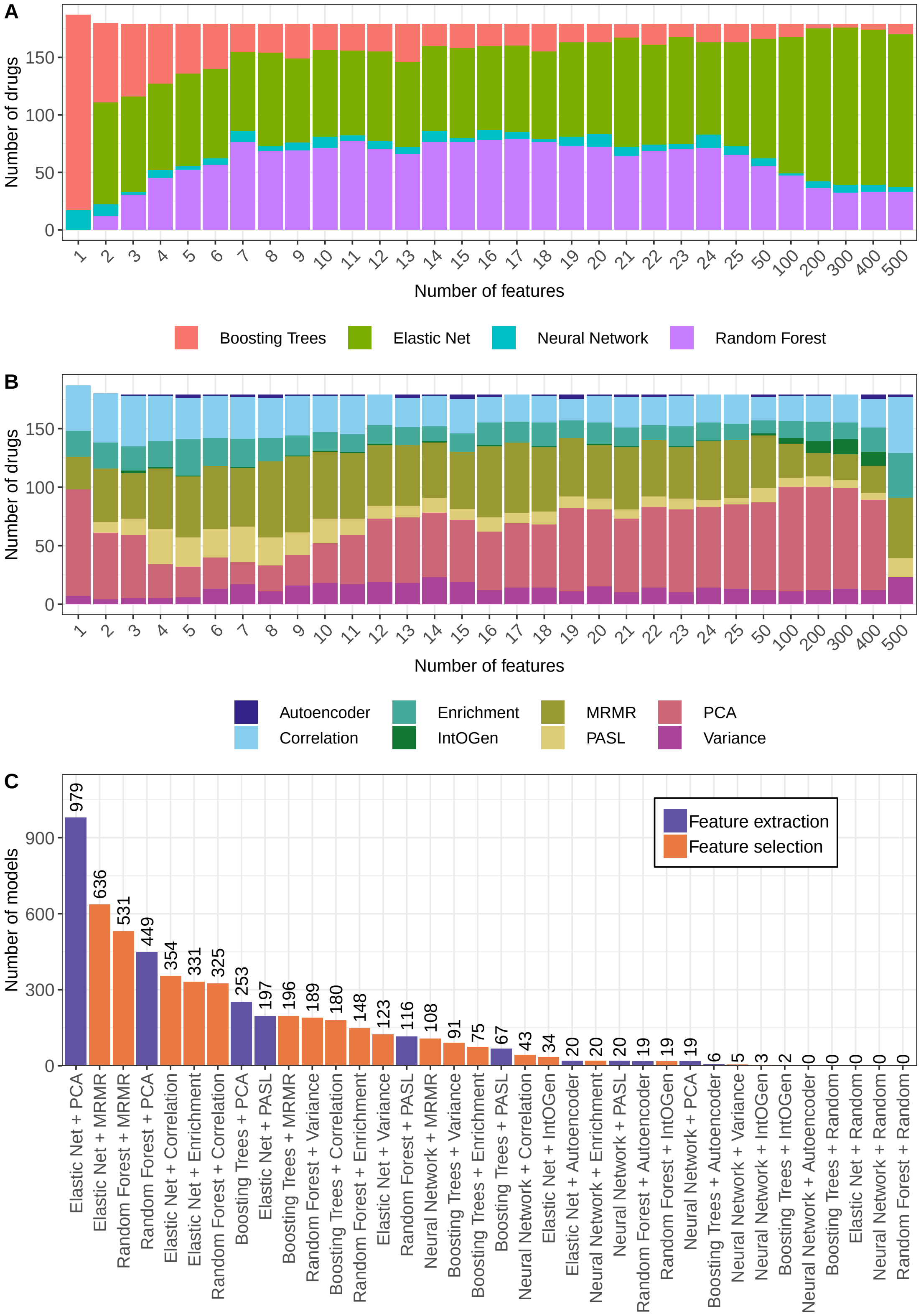


FIGURE 5.2: Best-performing ML algorithm and DR technique for each drug and number of input features. For each feature number k, Subfigure A shows for how many drugs each ML algorithm yielded the smallest test MSE. Subfigure B shows the analogous results for each DR algorithm. For $k = 1$, the height of the corresponding bar in both subfigures exceeds the number of drugs (179) since, for a few drugs, more than one ML-DR-combination resulted in the best performance. Thus, we show all best-performing combinations. For $k > 1$, this did not occur. Subfigure C shows how often a given combination of ML algorithm and DR method yielded the best results summarized over all drugs and k.

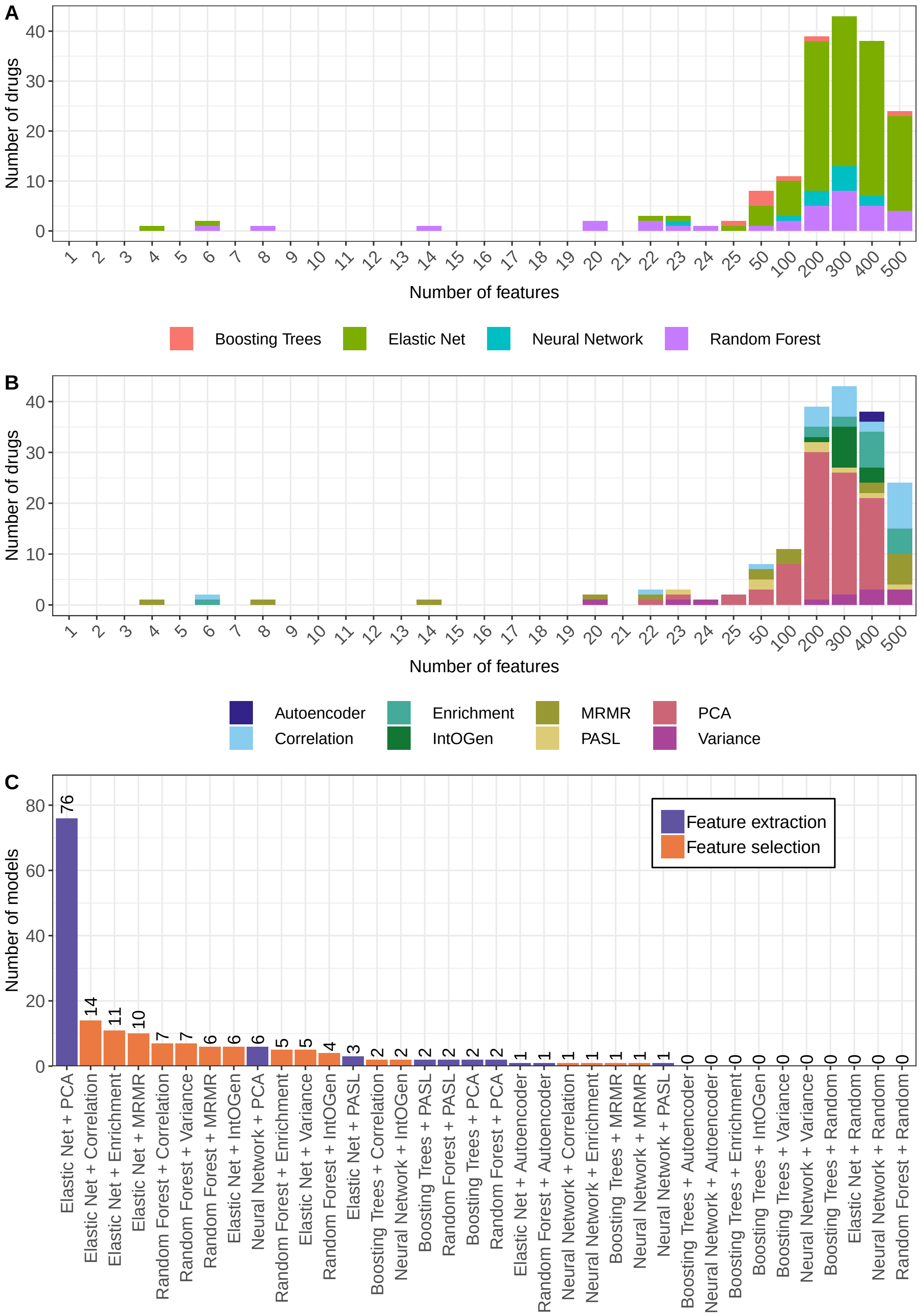


FIGURE 5.3: Best-performing ML algorithm, DR technique, and feature number for each drug. Subfigure A shows for how many drugs each ML algorithm yielded the smallest test MSE. For each drug, we only show the model for the k that resulted in the smallest test MSE. Subfigure B shows the analogous results for each DR algorithm. Subfigure C shows how often a given combination of ML algorithm and DR method yielded the best results summarized over all drugs.

As discussed previously, FE-based features are inherently difficult to interpret. Therefore, we performed the same analysis using FS algorithms only. The results are presented in Appendix Figures C.4 and C.5 and reveal that among all FS approaches, MRMR is chosen most often, followed by the correlation- and enrichment-based approaches. Note that all of these methods utilize not only the expression values but also the IC50 values to derive informative features.

5.2.3 Error Baseline and Trade-Offs

In this section, we systematically assess performance trade-offs between different models. To this end, we only consider the best-performing model for each drug and feature number k, i.e., the model resulting in the smallest test MSE. The MSE is frequently used to quantify prediction performance (cf. Section 4.3.1). However, it can be challenging determine whether the MSE obtained for a model is satisfactory without a proper baseline error. A straightforward approach to obtain a baseline is to use a *dummy model* that always predicts the mean response (here: mean ln(IC50)) of training samples. This model is based on the idea that – without any knowledge about informative features – predicting the mean response of the training data is the best guess to minimize the squared error (cf. Equation 4.4 in Chapter 4).

Figure 5.4A depicts the ratio between the test MSE of the best-performing model for each drug and k and the baseline MSE for the same drug. All models are an improvement over the baseline. For 80% of models, the improvement is at least 20% ($\frac{MSE}{Baseline} \leq 0.8$), and for 18% of models, the improvement is at least 40% ($\frac{MSE}{Baseline} \leq 0.6$).

In the same manner, errors of models with different input sizes can be compared: For most drugs, the best-performing feature number was around $k = 300$ (cf. Figure 5.3A), and smaller k resulted in larger errors (cf. Figure 5.1). In Figure 5.4B, we compare the test MSE of the best-performing model for each drug using $k = 300$ features to models with smaller k. For most models (63%), the MSE increase when using $k < 300$ features is rather small ($< 10\%$). Comparing models with $k = 300$ features to those using $k = 100$, only 3% of models show an MSE increase $> 10\%$. For $k = 15$, this number rises to 27%, but the model complexity is reduced drastically.

Our previous analyses also showed that PCA was the best-performing DR method (cf. Figure 5.3). However, features obtained through PCA are not easily interpretable. In contrast, the correlation-based FS is easy to interpret and implement. Thus, in Figure 5.4C, we compare the performance of the best-performing models for each drug and k using PCA- versus correlation-based features.

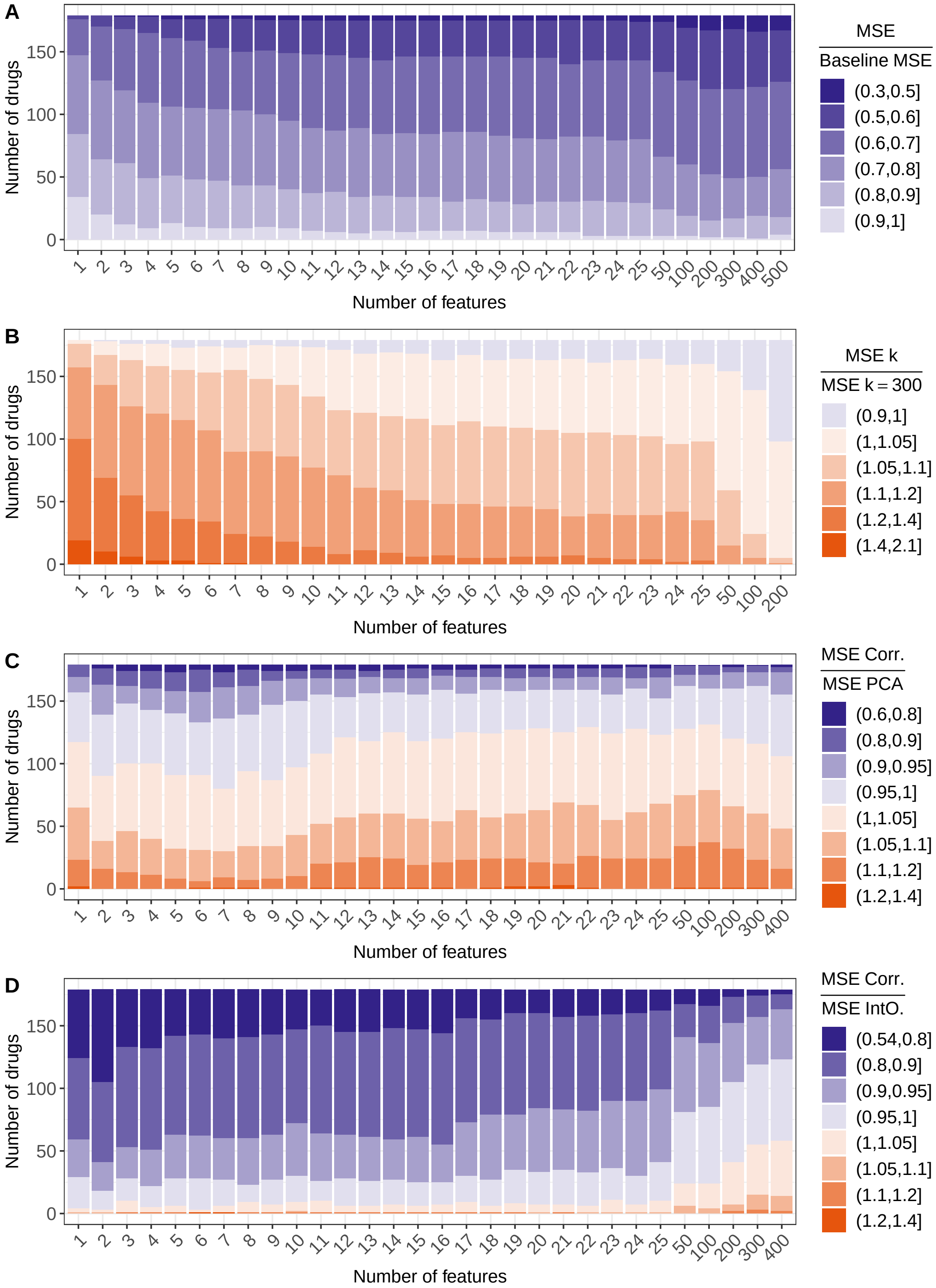


FIGURE 5.4: Performance comparison of different models, feature numbers, and feature types. Each subfigure depicts a comparison of test MSEs for different scenarios, where the MSE for one scenario is divided by the MSE of the other: Subfigure A compares the test MSE of the best-performing models for each drug and k to the MSE of drug-specific baseline models that always predict the mean ln(IC50) of the training samples. Subfigure B compares the test MSE of the best-performing models with $k < 300$ features to those with $k = 300$ features. Subfigures C and D compare the best-performing models for each drug and k using correlation-based features versus features derived from PCA or the IntOGen gene list, respectively.

Notably, for 37% of models, the correlation-based features outperform the PCA-based ones. For 52% of models, the increase in MSE from using the correlation-based FS instead of PCA is small ($< 10\%$). An increase of $> 20\%$ can only be observed for 0.4% of models. In Figure 5.4D, we analogously compare the correlation-based FS to the literature-based FS using the IntOGen cancer gene list. The correlation-based features outperform the literature-based ones for 93% of models. Among the 7% of models where the IntOGen features performed better, improvements were mostly small ($< 10\%$ in 94% of models). In summary, these results indicate that using the literature-based over correlation-based FS is not recommended. However, the improved interpretability of correlation-based features might warrant using this FS over PCA despite its slightly worse performance.

5.2.4 Improving Predictions for Sensitive Cell Lines

The MSE is a valuable means to assess the average model performance for previously unseen samples. However, one may also want to analyze how a model performs for specific groups of samples. In Chapter 1, we already highlighted the issue that sensitive cell lines are typically underrepresented in drug screening data, leading to poor prediction performance for these samples. Since cases of effective treatment are of particular interest for treatment recommendation, improving their predictions is highly desirable. In Chapter 6, we address this phenomenon explicitly by developing a novel ML approach called SAURON-RF. With SAURON-RF, we were able to improve predictions for sensitive cell lines by using – among other things – sample-specific weights. In the following, we show that predictions for sensitive cell lines can also be improved by using an error measure for hyperparameter tuning that increases the focus on accurate predictions for these samples. This idea is similar to sample-specific weights, where samples with higher weights have a greater impact on the error function during model training. However, the method discussed here does not alter the model training but only the selection of the best-performing hyperparameters afterwards. This approach can be beneficial when the training procedure cannot be modified, e.g., when models are built using a programming library that does not support sample weights or other training adjustments.

Conventionally, we would use the CV MSE as tuning measure, i.e., we select those hyperparameters for which the MSE averaged over all CV folds is smallest. The left side of Figure 5.5 depicts the distribution of test MSEs for all drugs when hyperparameters and settings for each drug are selected based on this measure. We separately evaluated the MSE for the subsets of sensitive and resistant cell lines from the test set. To identify samples as *sensitive* or *resistant*, we used the procedure by Knijnenburg et al., which is described in Section 3.2.4.2. As expected, the test MSE for sensitive cell lines is considerably larger than for resistant ones.

For the results shown on the right side of Figure 5.5, we did not use the conventional CV MSE to select hyperparameters and settings but instead calculated the CV MSE separately for the subsets of sensitive and resistant cell lines and averaged both values. Thus, errors for both subsets are weighted equally, even though the subset of sensitive cell lines contains much fewer samples. Using this tuning measure, the MSE for sensitive cell lines decreases considerably but remains larger than the MSE for resistant ones. Unfortunately, the error improvement for the sensitive samples comes at the cost of an error increase for the resistant samples. Depending on the application, this trade-off may still be warranted since the resistant samples are still predicted more accurately than the sensitive ones. However, increasing the error of resistant cell lines too much encompasses the risk of falsely predicting them as sensitive, i.e., overestimating treatment effectiveness. In Chapter 6, we show how a more favorable trade-off and even stronger improvements for the sensitive cell lines can be achieved using SAURON-RF.

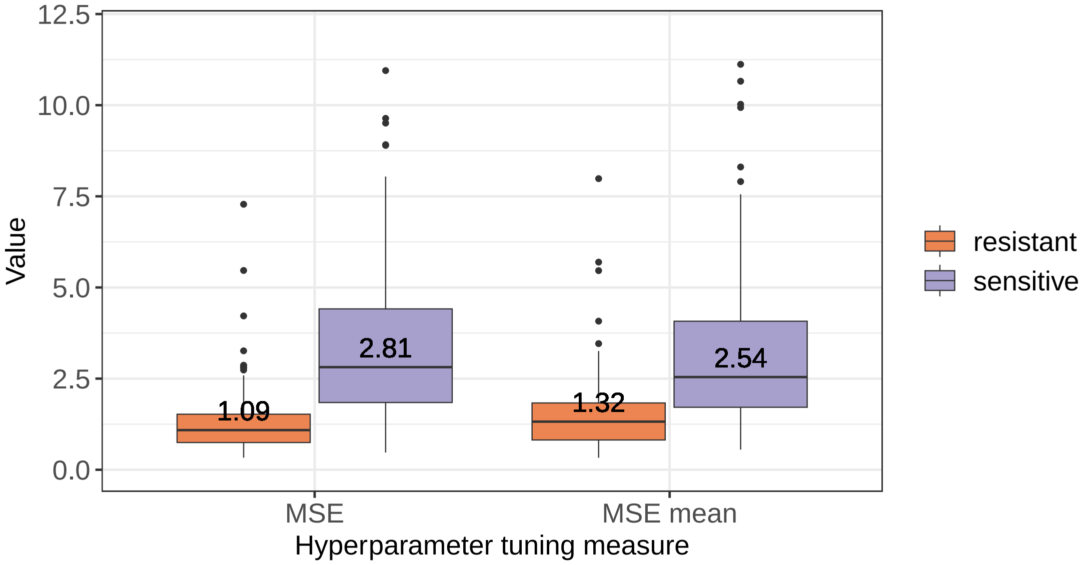


FIGURE 5.5: Effect of different hyperparameter tuning measures. This figure compares the test MSEs for sensitive and resistant cell lines when using two different measures to tune the hyperparameters and settings of each drug. On the left, the conventional CV MSE was used as tuning measure. On the right, we calculated the CV for the subsets of sensitive and resistant cell lines separately and then averaged both values. To identify a cell line as sensitive/resistant, we compare its ln(IC50) to a drug-specific threshold derived through a procedure described by Knijnenburg et al. [36]. Additionally, the median value for each box plot is shown.

5.2.5 Runtime

Next, we investigated the runtime of model training. Since a model generally needs to be trained only once, models with long runtime should still be preferred over models with shorter runtime if the performance of the former is superior. However, in a scenario where a large number of models should be trained, a high runtime can prevent the tuning of a large number of hyperparameter combinations. Boosting trees, random forests, and elastic nets were trained using 24 cores on an *Intel Xeon Gold 6248* (2.50GHz) CPU. Neural networks were initially trained using a *Nvidia Tesla V100-SXM2* GPU since GPUs are known to effectively parallelize computations needed in network training. However, for comparatively small networks, the overhead of transferring calculations to the GPU can outweigh the computational speedup, which we also observed in our experiments (see Appendix Figure C.6). Hence, we switched from GPU to CPU (*Intel Xeon E5-2698 v4*, 2.20GHz, 24 cores) for neural networks.

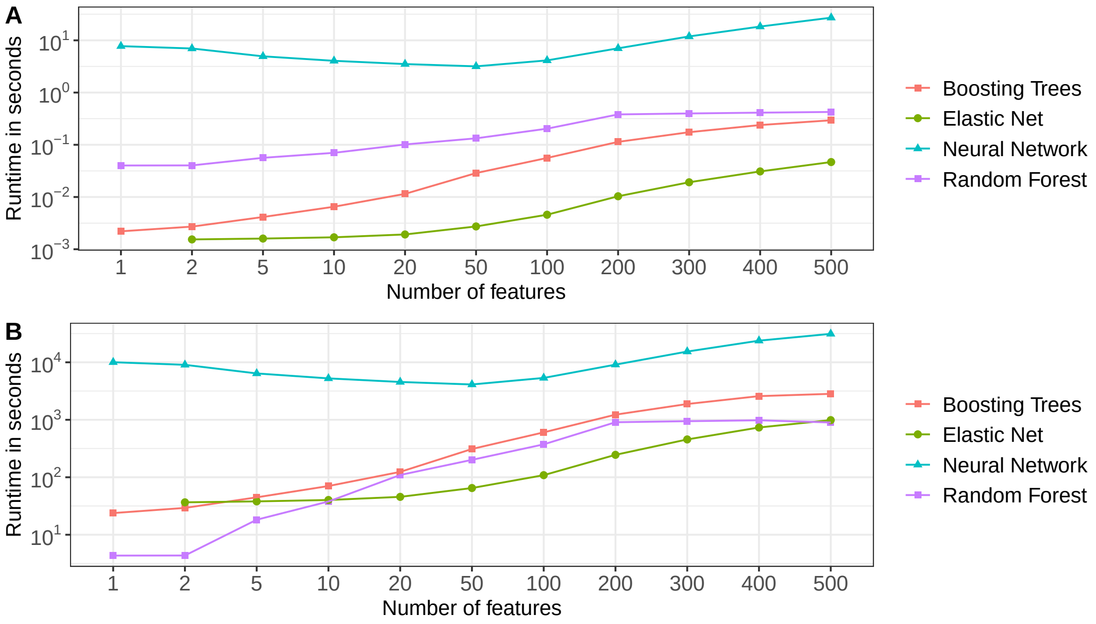


FIGURE 5.6: Runtime comparison of ML algorithms. Subfigure A depicts the mean training duration for a single model (i.e., one hyperparameter combination) for increasing feature numbers. Subfigure B depicts the duration of performing the 5-fold CV plus fitting of models on the whole training data accumulated over all hyperparameter combinations.

Figure 5.6A depicts the duration of training a single model (i.e., one hyperparameter combination) using each ML algorithm averaged over all trained models. The runtime of neural networks is considerably larger than that of the other algorithms, with elastic net being the fastest. Since we used a different CPU to train neural networks compared to the other algorithms, the runtime comparisons may be influenced by hardware disparities.

However, even when comparing neural networks to the second slowest algorithm, i.e., random forests, the average runtime of neural networks is 18 times ($k = 200$) to 193 times ($k = 1$) longer. Compared to elastic net, the runtime of neural networks is increased by a factor of over 580 for each k. These differences are likely not attributable to the hardware differences alone.

Figure 5.6B depicts the combined runtime of performing the complete CV and training of final models on all hyperparameter combinations for a given setting. Although we tested the most hyperparameter combinations for elastic nets (cf. Table 5.2), they remain the fastest approach for $k > 10$. Neural networks remain the slowest, even though we investigated the fewest hyperparameter combinations.

5.2.6 Model and Feature Interpretability

Drug sensitivity prediction aims to provide accurate models but also to identify markers of treatment sensitivity/resistance. However, not all models are equally interpretable. As discussed above, inputs obtained through FS are more interpretable than those obtained through FE. Additionally, ML algorithms exhibit great differences regarding their inherent interpretability. In Table 5.3, we assess the interpretability of the four investigated ML algorithms based on model *transparency* and *explainability* as introduced in Section 4.3.3. Briefly recapitulated, transparency denotes the inherent understandability of a model, while explainability involves using post-hoc explanations to understand a model's predictions, e.g., through feature importance scores.

However, training an ML model is not always necessary to identify features that impact drug response: Three of the investigated FS approaches (correlation-based, enrichment-based, MRMR) not only consider the expression data but also the drug response values. Consequently, the chosen features may already provide information on potential markers of sensitivity/resistance. The selected features can not only be investigated for each drug individually but also across drugs.

Since we found that MRMR is the best-performing FS method, we investigated the selected features more closely: Figure 5.7A depicts how often each gene was chosen by the MRMR FS for $k = 25$ across all drugs and its average rank in the drug-specific MRMR lists. Features that are chosen often, such as BCL2L1 and SLC27A5, might be interesting biomarkers across drugs. Indeed, BCL2L1 expression was shown to prevent apoptosis, thereby conferring multi-drug resistance to cancer cell lines [349]. Increased apoptosis and drug sensitivity were observed when BCL2L1 was silenced or inhibited [350–352]. Gao et al. found that SLC27A5 deficiency was related to poor prognosis, proliferation, and drug resistance in hepatocellular carcinoma [353]. Knockout of SLC27A5 activates the KEAP1/NRF2 pathway [353], which is linked to chemoresistance in non-small cell

Table 5.3: Interpretability of ML models. We assess model interpretability based on the concepts of *transparency* and *explainability* as described in Section 4.3.3. We only list the explainability methods that are readily available in the used R/Python packages (cf. Table 5.2).

Model	Transparency	Explainability
Elastic net	+ easily interpretable feature coefficients	• feature importance: absolute value of feature coefficients; sign of coefficient denotes impact direction
Random forest	+ easily interpretable decision splits − typically large number of trees	• feature importance: error improvement obtained from splits using certain feature; error increase when feature is randomly perturbed • samples similar to given input: training samples reaching same leaf nodes
Boosting trees	+ easily interpretable decision splits − typically large of trees − trees affect predictions to varying degrees	• feature importance: error improvement obtained from splits using certain feature; error increase when feature is randomly perturbed
Neural network	− typically thousands of parameters − complex, multi-layered computations	

lung cancer patients [354] and different types of cancer cell lines [355]. In contrast to such multi-drug markers, features rarely chosen by MRMR but with a small rank are likely important for specific drugs only and can be assessed to study drug-specific mechanisms of treatment response.

We next investigated whether the features that are selected often by MRMR are also the ones with the highest impact in elastic net models trained with those features. We chose elastic nets for this analysis due to the straightforward interpretation of their coefficients (cf. β in Equation 4.12). Figure 5.7B depicts how often the selected MRMR features had a non-zero contribution in the trained elastic nets and their mean rank based on the absolute value of assigned coefficients. Again, BCL2L1 and SLC27A5 were relevant features for a large number of drugs (55 and 46 drugs, each). The sign of the coefficients indicates whether increasing the expression of a gene positively or negatively affects the predicted IC50s. For 98% of genes, their contribution was either always positive or always negative for at least 99% of drugs, indicating that features generally have a consistent impact on drug response.

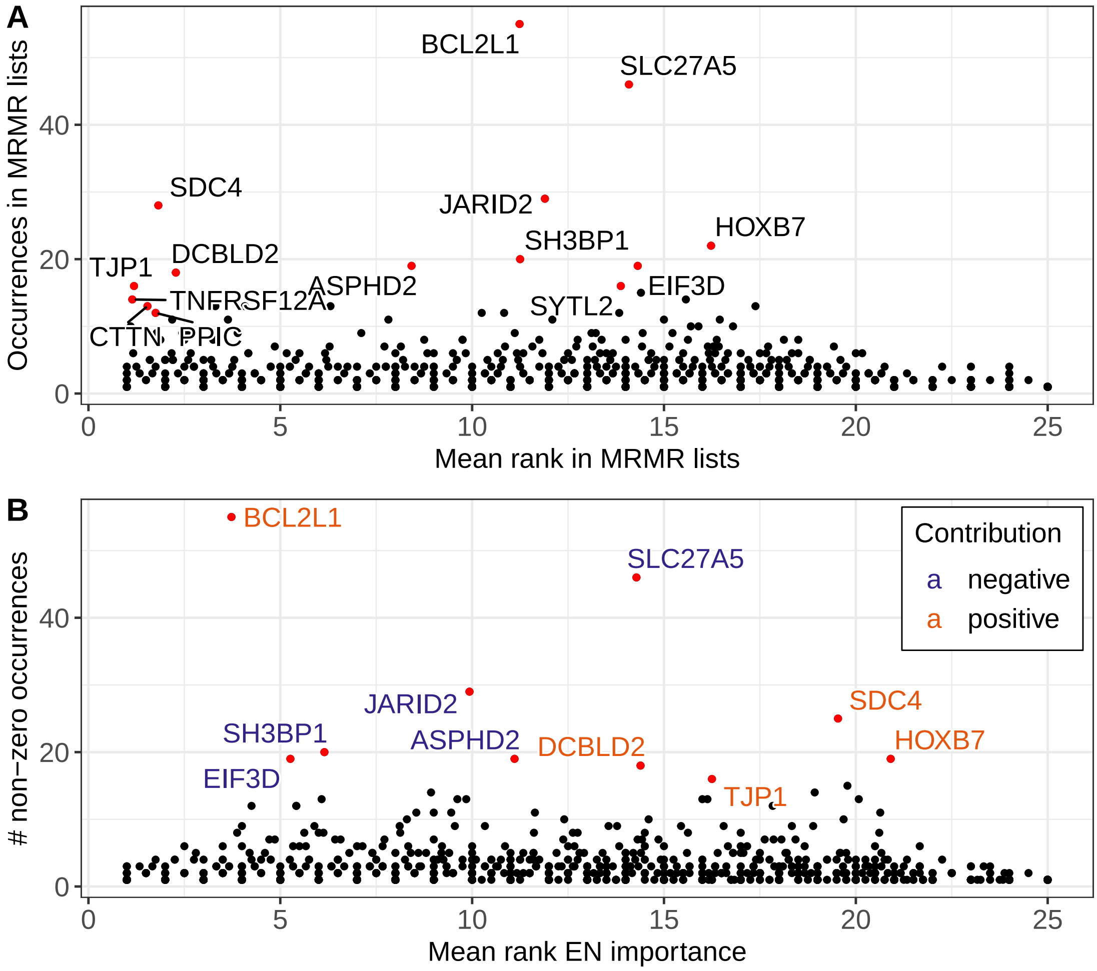


FIGURE 5.7: Feature importance of MRMR features. Subfigure A shows the importance of features selected by the MRMR FS with $k = 25$ for all drugs. The x-axis denotes the mean rank of each gene in the feature lists of all drugs, and the y-axis denotes the number of drug lists in which the gene occurs. Subfigure B shows the importance of the MRMR features in the best-performing elastic net (EN) models trained using these features. The x-axis denotes the mean rank of each gene according to the absolute value of feature coefficients derived from the trained models, and the y-axis denotes the number of models in which the feature had a non-zero coefficient. The color denotes whether the genes have a negative/positive coefficient (cf. β_j in Equation 4.12) in the trained models, indicating their tendency to de-/increase predicted IC50s.

5.2.7 Impact of DR on a Multi-Omics Multi-Drug Deep Learning Model

Our benchmarking mainly focuses on predicting drug responses based on gene expression features using standard ML algorithms. However, state-of-the-art prediction models exhibit increasingly complex architectures and are often based on multi-omics characterizations of cell lines, sometimes including also molecular characterizations of drugs [356]. To highlight that even complex models can benefit from using different DR methods, we investigated the effect of different DR methods on a multi-omics multi-drug deep learning model by Chiu et al. [292] and compared its performance to that of elastic nets and random forests. Their approach uses gene expression and mutation data as input, which are projected into a lower dimension ($k = 64$, each) using autoencoders. The encoders are connected to a deep neural network, with drug-specific output nodes predicting each drug's ln(IC50). To conform to our analysis setup, we slightly modified the autoencoder pre-training. For an in-depth description of the model and our performed analyses, please refer to Appendix C.2. The results can be summarized as follows:

- The model by Chiu et al. is outperformed by both drug-specific elastic net and random forest models using $k = 64$ correlation-based expression features for all 170 investigated drugs (cf. Appendix Figure C.9A,B).
- Even when the autoencoders by Chiu et al. are trained to generate larger feature embeddings of as many as $k = 500$ features for each omics type, they are outperformed by drug-specific elastic net and random forest models using only $k \leq 10$ features for 95% and 90% of drugs, respectively (cf. Appendix Figure C.8).
- When replacing the autoencoders in Chiu et al.'s approach with PCA or correlation-based features, the performance is improved for 83% and 76% of drug and k combinations, respectively (cf. Appendix Figures C.9C,D).
- Using solely expression features was superior to using solely mutation features for a large majority of the investigated models and feature numbers (cf. Appendix Figure C.10).

In summary, this exemplary analysis highlights that single-drug models with small feature numbers can outperform more complex multi-drug and multi-omics approaches. It also proves again that the choice of ML/DR method substantially impacts predictions.

5.3 Discussion

We performed a comprehensive analysis of the prediction of IC50 values using four machine learning (ML) algorithms in combination with six feature selection (FS) and three feature extraction (FE) techniques. Our evaluations on the GDSC2 dataset show that elastic nets using features obtained through PCA yielded the smallest test MSE for 76 of 179 investigated drugs. Elastic nets also showed the lowest runtime and allow for a straightforward identification of features with a strong impact on predictions. In contrast, neural networks, including the more sophisticated deep learning approach by Chiu et al. [292], had the worst performance. This aligns with the observations by Li et al. [24] and Chen and Zhang [25, 243] who also found that neural networks, despite their recent popularity for drug sensitivity prediction, are frequently outperformed or at least matched by other ML approaches for this task.

However, in our analyses, the large runtime of neural networks only permitted us to perform a relatively limited hyperparameter tuning. Thus, more extensive tuning might improve the performance of neural networks further. Furthermore, the runtime comparison of neural networks to the other algorithms is impaired by the use of different CPUs.

Among the FS methods, the MRMR-based approach performed best. In general, FS methods that consider the drug response performed better than methods using only expression values. Methods that do not consider either drug response or gene expression, like the literature-based or random FS, performed worst. However, on a dataset other than the GDSC, features like known cancer genes might yield more robust predictions than features tailored to a specific dataset. Additionally, our literature-based FS uses the same features for all drugs. Thus, considering also drug-specific response markers from literature might improve performance [37].

It also remains to be investigated how models derived from cell lines translate to xenografts and humans, where drug absorption, availability at the target site, and side effects are crucial factors that cannot be fully captured by cell lines (cf. Section 3.1). For this application, models would likely benefit from additional features that account for such mechanisms, which would, however, need to be derived from (often scarce) xenograft/human data.

Regarding the investigated omics types, our analyses focused mainly on gene expression data since it is the most informative data type for drug sensitivity prediction [23, 176] and our analyses using mutation data agree with these findings. However, combining different omics types might prove beneficial. There exist also several ML approaches that employ not only cell line features but also drug features in the form of molecular fingerprints [356] (see also Chapter 8). Here, FS could provide insight into which drug properties impact treatment response. Additionally, in an attempt to make models more interpretable, biological mechanisms are often explicitly encoded into prediction models, e.g., by using pathway-layers in neural networks [24, 294, 357] or exploiting known protein interactions [177]. Li et al. found, however, that the explicit incorporation of biological knowledge may decrease model performance and lead to false conclusions [24]. Hence, the assumptions introduced by adding such knowledge to a model should be carefully investigated.

Our benchmarking primarily focused on training models for each drug separately. However, multi-drug models such as the multi-task network by Chiu et al. [292] enjoy popularity. Our analyses show that single-drug models can outperform multi-drug models and might, thus, be preferred when predicting the drug response for an unknown sample (cell line) to a known drug. In contrast, for an unknown drug, single-drug models are not directly applicable, whereas some multi-drug models can be applied given that a representation of the drug of interest, e.g., a molecular fingerprint, is available (cf. models trained in Chapter 8).

Besides the various performance evaluations, we also showcased how to assess the quality of a model by comparing its performance to a simple baseline model and how to assess the trade-off between performance and interpretability when using (1) fewer features and (2) FS compared to FE. Interpretability and trust in predictions are crucial when ML models should eventually be used for clinical decision support. However, interpretability often suffers as models become increasingly complex, which warrants the question of how complex a model truly needs to be to generate sufficiently accurate predictions. Our performance comparisons with the deep learning approach by Chiu et al. [292] showed that interpretable models with small feature numbers can substantially outperform complex prediction algorithms. Moreover, our results indicate that complex models equally benefit from using simple DR methods.

Overall, we are convinced that the methods and evaluation strategies discussed here are helpful tools for developing and assessing ML models for drug sensitivity prediction, independent of the specific algorithms and features at hand.

Chapter 6

SAURON-RF: Improving Predictions for Drug-Sensitive Cell Lines

A multitude of machine learning (ML) approaches have been developed to predict the drug response of cancer cell lines based on their molecular characteristics. While classification approaches typically focus on distinguishing between *sensitive* and *resistant* samples, regression approaches predict a continuous response measure such as the IC50 value that quantifies the degree of sensitivity. In this chapter, we address a crucial challenge of ML-based drug sensitivity prediction: making accurate predictions for drug-sensitive cell lines. Those cell lines that react sensitively to a drug are of particular interest since they represent cases of potential treatment success, and our final goal is to identify those drugs that result in the most effective treatment for a given cell line (patient). However, sensitive samples are heavily underrepresented in drug screening data since most targeted anti-cancer drugs act highly specific. In the GDSC database, for example, the average sensitive-to-resistant ratio of cell lines per drug is only 1:10 [37], which negatively impacts the performance of ML models trained on this data for the underrepresented samples.

The phenomenon that one response class in a classification problem is underrepresented is known as *class imbalance* [358]. It leads classifiers trained on such imbalanced data to frequently misclassify the underrepresented class. Even though sensitive cases are of particular interest for personalized medicine, there are relatively few approaches for drug sensitivity prediction that attempt to counteract class imbalance (cf. Table 6.1): LOBICO by Knijnenburg et al. [36] and MERIDA by Lenhof et al. [37] are integer linear programs where sample-specific weights are employed to enhance the importance of the most sensitive cell lines. MOLI by Sharifi-Noghabi et al. is a deep neural network where the importance of sensitive samples is increased through upsampling [314]. The multi-drug deep neural network RAMP by Lee et al. employs a network-based feature embedding and a *contrastive regularization* technique, which both aim to distinguish sensitive from resistant cell lines to enhance the prediction quality for both groups [315]. Even though these approaches can mitigate class imbalance to some degree, an accurate classification remains challenging.

Similar to class imbalance, there exists a comparable challenge for regression, known as *regression imbalance* [359]: While responses around the mean of the training data can be predicted accurately, the lower (here: most sensitive) and upper (most resistant) ends of the response scale are systematically over-/underpredicted. The underrepresentation of sensitive cell lines further aggravates this problem since the mean of the distribution is shifted towards the resistant samples. Consequently, the most sensitive cell lines are often predicted the least accurately.

Compared to class imbalance, regression imbalance is even less discussed in drug sensitivity prediction literature (cf. Table 6.1). Currently, we are only aware of one approach that explicitly addresses it, namely RWEN by Basu et al. [38]. They train an elastic net with sample weights (cf. Section 4.2.1), where the weights for the most sensitive samples are iteratively adapted to enhance their predictions.

Another approach that can alleviate the problem of regression imbalance (even though it was not explicitly designed for this task) is HARF (*Heterogeneity Aware Random Forests*) by Rahman et al. [285]: They train a conventional regression random forest for the prediction of continuous drug responses but integrate information on the cancer type of each cell line (e.g., *skin*, *lung*, *breast*) into the model. Consequently, the model can predict the cancer type of previously unseen cell lines. Next, this information is used to adapt the regression prediction through tree-specific weights. Thereby, Rahman et al. generate predictions around the mean response of each cancer type. If the considered cancer types have sufficiently different drug responses, this approach mitigates the problem of regression imbalance. In their analyses, Rahman et al. only focus on training models for two cancer types with distinct responses and show that this approach outperforms several other ML approaches [285]. However, limiting the considered cancer types discards a large portion of the available samples, which aggravates the problem of data scarcity (cf. Section 4.3.4). Furthermore, for cancer types without significant differences in drug response, the regression imbalance should persist.

In this chapter, we present our prediction approach SAURON-RF (*SimultAneoUs Regression and classificatiON Random Forests*), which is based on HARF but has three major modifications that are designed to counteract both class and regression imbalance:

1. Instead of cancer types, we use *sensitive* and *resistant* drug response as classes.

2. We increase the importance of sensitive cell lines through upsampling or sample-specific weights.

3. We investigate different tree-weighting approaches to improve the regression predictions further.

Additionally, our approach can perform both a classification and a regression simultaneously, making it well-suited for drug prioritization. The goal of drug prioritization is to first identify those drugs that are effective for a given cell line via classification and to subsequently rank those drugs by their effectiveness via regression. We will discuss drug prioritization in detail in Chapter 7.

In the following, we present the HARF and SAURON-RF algorithms in detail. Using data from the GDSC database, we then showcase how class and regression imbalance affect various ML algorithms and how these issues can be alleviated using the different modifications of SAURON-RF. Our approach outperforms various classification and regression random forests, including HARF, and strongly improves predictions for the sensitive cell lines at a comparatively smaller loss in performance for the resistant ones.

Author Contributions

The content of this chapter is based on the following publication:

Lenhof, K., Eckhart, L., Gerstner, N., Kehl, T., & Lenhof, H. P. (2022). **Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method.** Scientific Reports, 12(1), 13458. DOI: 10.1038/s41598-022-17609-x

The idea for SAURON-RF was conceived by Kerstin Lenhof. Kerstin Lenhof, Hans-Peter Lenhof, and I designed the study based on the study design and findings from my Master's thesis [116]. Kerstin Lenhof implemented the SAURON-RF algorithm, while I implemented the feature selection and the ML models other than SAURON-RF. The presented analyses were conducted by Kerstin Lenhof and myself. Kerstin Lenhof also drafted the publication manuscript, which I edited and reviewed. All publication authors discussed the results and reviewed the manuscript.

Table 6.1: Overview on the consideration of class and regression imbalance in drug sensitivity prediction literature. For 28 publications, this table lists whether class and regression imbalance have been addressed in the respective approach. The row-coloring corresponds to the type of supervised learning that was employed (blue: classification, white: regression, yellow: classification and regression).

Publication	Class imbalance counteracted	Regression imbalance counteracted
LOBICO (2016) [36]	✓	(✓)*
Stanfield et al. (2017) [177]	✗	✗
Deep-Resp-Forest (2019) [179]	✗	✗
MOLI (2019) [314]	✓	✗
MERIDA (2021) [37]	✓	(✓)*
GraphCDR (2021) [321]	✗	✗
RAMP (2022) [315]	✓	✗
KBMTL (2014) [301]	✗	✗
SAURON-RF (2022) [39]	✓	✓
Menden et al. (2013) [282]	✗	✗
Zhang et al. (2015) [283]	✗	✗
SRMF (2017) [284]	✗	✗
HARF (2017) [285]	✗	(✓)†
HNMDRP (2018) [307]	✗	✗
Matlock et al. (2018) [286]	✗	(✓)‡
RWEN (2018) [38]	✗	✓
CDRscan (2018) [289]	✗	✗
QRF (2018) [290]	✗	✗
NCFGER (2018) [325]	✗	✗
DeepDR (2019) [292]	✗	✗
NetBiTE (2019) [293]	✗	✗
Deng et al. (2020) [294]	✗	✗
ADRML (2020) [296]	✗	✗
PathDSP (2021) [297]	✗	✗
GraphDRP (2022) [299]	✗	✗
NeRD (2022) [322]	✗	✗
GADRP (2023) [323]	✗	✗
mVAEN (2023) [324]	✗	✗

* While LOBICO and MERIDA do not explicitly address regression imbalance, they employ a sample weighting scheme where cell lines at the lower/upper end of the sensitivity scale are weighted most heavily. Such weights are, in principle, suited to counteract regression imbalance.

† While HARF was not explicitly designed to counteract regression imbalance, the approach may counteract regression imbalance in cases where the chosen cancer types differ notably in their drug response.

‡ While the approach by Matlock et el. was not explicitly designed to counteract regression imbalance, they show that it, nevertheless, improves predictions for sensitive cell lines compared to conventional ML models.

6.1 Materials and Methods

In the following, we first describe the dataset we employed to perform the analyses in this chapter. Next, we discuss the in- and outputs of the used ML models. Afterward, we present the details of the HARF algorithm by Rahman et al. [285] and our SAURON-RF algorithm.

6.1.1 Dataset

Data for our analyses was obtained from the GDSC database (Release 8.3), which we described in detail in Section 3.4. More specifically, we downloaded normalized expression values for 17,419 genes and drug screening data in the form of logarithmized IC50 values for all 198 drugs screened in the GDSC2 dataset. Out of these 198 drugs, we only considered those 86 drugs for which sensitivity measures for at least 750 cell lines are provided (cf. Appendix Table C.1).

To obtain discrete drug responses, we discretized the continuous IC50 values by comparing them to a drug-specific threshold t. The thresholds were derived using a method by Knijnenburg et al. [36] described in Section 3.2.4.2. Cell lines with $ln(IC50) \leq t$ are considered *sensitive*. Otherwise, they are considered *resistant*.

6.1.2 Model In- and Outputs

We trained drug-specific regression/classification models using four conventional ML approaches (elastic net, boosting trees, random forests, neural networks, cf. Section 4.2), as well as our novel prediction approach SAURON-RF and the HARF algorithm by Rahman et al. [285], which will both be described below. Each model is trained to predict the continuous/discrete drug response of a cell line based on its gene expression. As input features, we used $k = 20$ drug-specific genes determined through a heuristic feature selection by Kwak and Choi [339], which is based on the minimum-redundancy-maximum-relevance (MRMR) principle [338]. In the previous chapter, i.e., our drug sensitivity benchmarking, we found that this heuristic was one of the best-performing dimension reduction methods. While the details of the algorithm can be found in Section 5.1.4, the algorithm can briefly be summarized as selecting features that have a strong dependence on the response (i.e., high *relevance*) but a weak dependence between each other (i.e., small *redundance*).

6.1.3 Model Training and Testing

For each drug, we divided the available cell lines into a training set (80% of cell lines) and a test set (20%). On the training set, we performed a 5-fold cross-validation (CV) to determine the best-performing hyperparameters of the different algorithms (see Appendix Table D.1). For the analyses presented in Section 6.2.3, we additionally treated the number of input features k as a hyperparameter and optimized $k \in \{20, 40, 60, 80, 100\}$. As error measure for hyperparameter tuning, we used the mean squared error (MSE, cf. Equation 4.65) for regression models (including HARF and SAURON-RF), and Matthew's correlation coefficient (MCC, cf. Equation 4.72) for classification models. With the determined hyperparameters, one final model is trained on the complete training data, and its performance is evaluated on the test set.

6.1.4 Heterogeneity Aware Random Forests

Heterogeneity Aware Random Forests (HARF) by Rahman et al. [285] is an algorithm for the prediction of a cell line's drug response based on its gene expression. HARF is based on a regression random forest (RF) as described in Section 4.2.4.2. However, the forest is extended with class information on the cancer type (e.g., *skin*, *lung*, *breast*) of each training cell line in the leaf nodes. Consequently, the trained regression RF can predict the cancer type of an unseen sample, similar to a classification RF (cf. Section 4.2.4.2).

Briefly summarized, the core idea of HARF is to predict the cancer type of a sample and to utilize this information when predicting the sample's drug response. More precisely, the HARF algorithm starts by fitting a conventional regression random forest for the prediction of continuous drug response values as described in Section 4.2.4.2. Now let $C = \{c_1, \dots, c_K\}$ be a set of K different classes, which in the HARF algorithm correspond to cancer types. Each cell line in the training data is assigned to one class. Consequently, the trained regression forest can be used to predict the cancer type of a sample x, just like a classification random forest: Each tree predicts the cancer type of x as the majority class of cancer types in the reached leaf node, and the prediction for the whole forest corresponds to the majority vote over all trees.

Next, HARF employs this class prediction to adapt the prediction of the continuous drug response: Recall that a conventional regression random forest makes a prediction $\hat{f}(x)$ for sample x by averaging the tree-specific predictions $\hat{f}_b(x)$ for each tree $b \in \{1, ..., B\}$:

$$\hat{f}(x) = \sum_{b=1}^{B} w_b(x) \cdot \hat{f}_b(x) \tag{6.1}$$

Here, $w_b(x) = \frac{1}{B}$ denotes the weight of tree b for the prediction of sample x. HARF modifies these tree-specific weights as follows:

$$w_b(x) = \frac{I_b(x)}{\sum_{\beta=1}^{B} I_\beta(x)} \tag{6.2}$$

The indicator function $I_b(x)$ is 1 if the class prediction of tree b for sample x is equal to the class prediction of the whole forest and 0 otherwise. Thus, HARF only considers a subset of all trees in the forest for the regression prediction, namely those trees where the class prediction agrees with the majority vote.

6.1.5 Simultaneous Regression and Classification Random Forests

Our approach, SAURON-RF, is based on the idea of HARF to augment a regression random forest with class information. However, we propose three modifications to HARF that aim to improve predictions for the group of sensitive cell lines:

1. Instead of cancer types, we use *sensitive* and *resistant* drug response as classes.
2. We increase the importance of sensitive cell lines through upsampling or sample-specific weights.
3. We adapt the binary tree weights proposed by HARF (cf. Equation 6.2).

In the following, we describe each of these modifications in detail. An overview of the entire SAURON-RF workflow is shown in Figure 6.1.

6.1.5.1 Class Definition

In SAURON-RF, the used class information does not correspond to cancer types as in HARF but instead denotes whether a cell line is *sensitive* or *resistant* to the drug of interest, i.e., $C = \{sensitive, resistant\}$. While we only consider a binary classification in this chapter, our algorithm can be extended to an arbitrary number of classes (cf. Chapter 7 where we add a third class to SAURON-RF denoting an *intermediate* drug response).

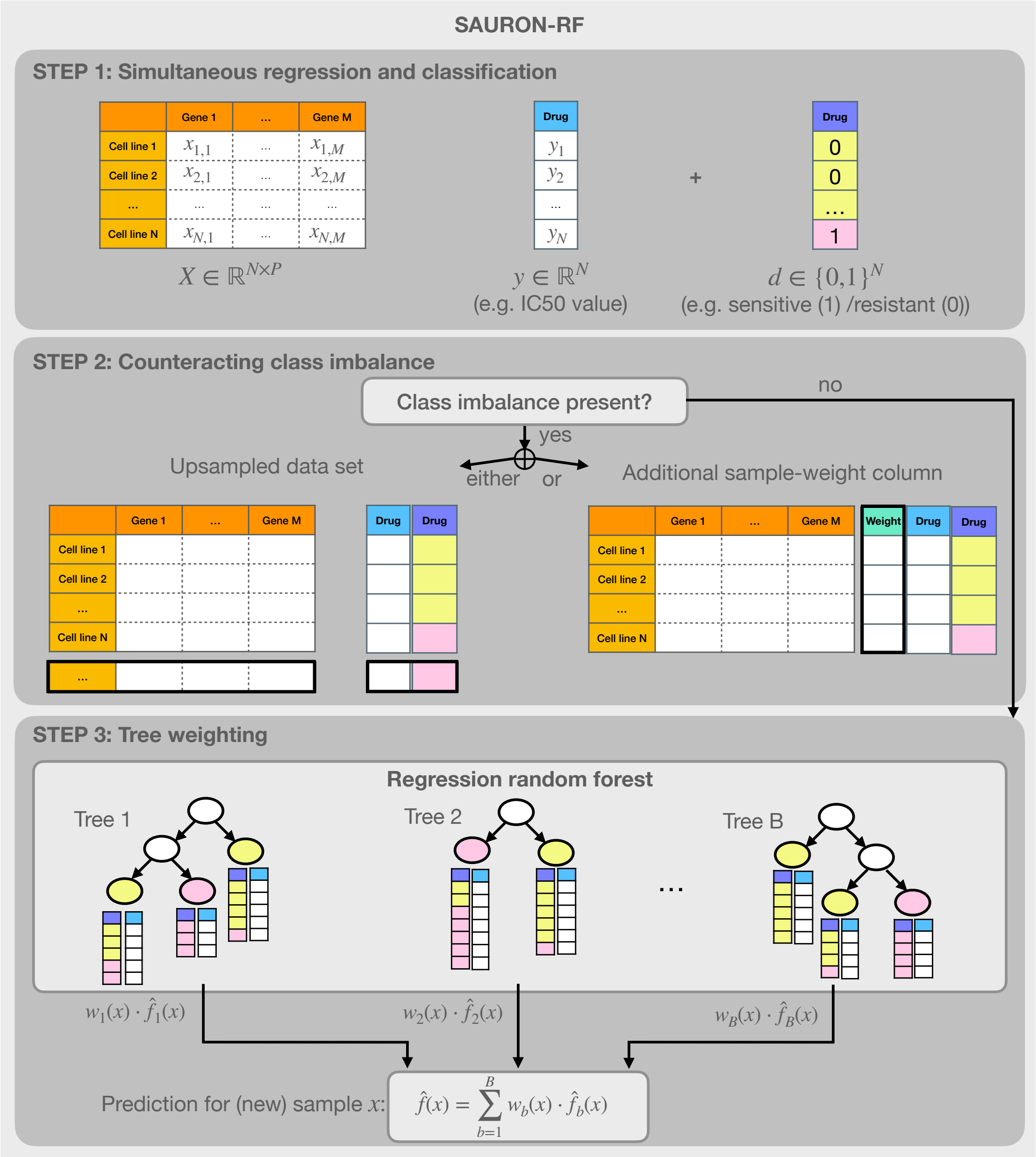

SAURON-RF
STEP 1: Simultaneous regression and classification
Gene 1
...
Gene M
Cell line 1
Cell line 2
...
Cell line N
$x_{1,1}$
$x_{1,M}$
$x_{2,1}$
$x_{2,M}$
$x_{N,1}$
$x_{N,M}$
$X \in \mathbb{R}^{N \times P}$
Drug
y_1
y_2
y_N
$y \in \mathbb{R}^N$
(e.g. IC50 value)
+
Drug
0
0
...
1
$d \in \{0,1\}^N$
(e.g. sensitive (1) /resistant (0))
STEP 2: Counteracting class imbalance
Class imbalance present?
no
yes
either
or
Upsampled data set
Additional sample-weight column
Weight
Drug
STEP 3: Tree weighting
Regression random forest
Tree 1
Tree 2
Tree B
...
$w_1(x) \cdot \hat{f}_1(x)$
$w_2(x) \cdot \hat{f}_2(x)$
$w_B(x) \cdot \hat{f}_B(x)$
Prediction for (new) sample x:
$\hat{f}(x) = \sum_{b=1}^{B} w_b(x) \cdot \hat{f}_b(x)$

Figure 6.1: Workflow of SAURON-RF. This figure summarizes the three steps of the SAURON-RF algorithm for generating one drug-specific prediction model. In Step 1, we perform a feature selection to reduce the dimensionality of the input gene expression matrix. Additionally, the discrete drug response is derived from the continuous IC50 values. In Step 2, class imbalance can be counteracted by either upsampling the minority class or by using sample-specific weights. In Step 3, a regression random forest is trained to predict the ln(IC50) value of a cell line based on its expression of the genes selected in Step 1. Next, each leaf sample in the trained forest is annotated with its discrete drug response. Consequently, the forest can now be used to classify a new sample x' as *sensitive* or *resistant*, just like a conventional classification random forest. The continuous prediction for x' is obtained by multiplying the tree-specific predictions $\hat{f}_b(x')$ with the tree-specific weights $w_b(x')$, which in some cases depend on the classification of x'.

6.1.5.2 Sample-Specific Weights and Upsampling

We investigated several techniques for increasing the importance of sensitive samples through sample-specific weights or upsampling. Both sample weights and upsampling affect the random forest in two ways: (1) the splitting of nodes and (2) the prediction for a new sample, which will be detailed below.

Simple sample weights: For this approach, we determine weights such that the sum of weights for the minority class (here: sensitive cell lines) is equal to the sum of weights for the majority class (here: resistant cell lines). The weight for training sample i is given by:

$$w_i^* = \begin{cases} 1, & \text{if sample } i \text{ belongs to the majority class} \\ \frac{N_{\text{Maj}}}{N_{\text{Min}}}, & \text{if sample } i \text{ belongs to the minority class} \end{cases} \tag{6.3}$$

Here, N_{Maj} and N_{Min} denote the number of samples belonging to the majority and minority class, respectively.

Linear and quadratic sample weights: For this weighting, we employ the drug-specific discretization threshold t that was used to classify samples as sensitive ($ln(IC50) \leq t$) or resistant ($ln(IC50) > t$). The idea is to assign higher weights to samples the further their drug response is away from t, thereby increasing the importance of the most sensitive/resistant samples. This approach was originally introduced by Knijnenburg et al. [36] and later adapted by Lenhof et al. [37]:

$$w_i^* = \frac{|y_i - t|^g}{2 \cdot \sum_{n=1}^{N} I(d_n = d_i) \cdot |y_n - t|^g} \tag{6.4}$$

The numerator measures the distance of the continuous drug response y_i of sample x_i from the threshold t. The exponent $g \in \{1, 2\}$ determines whether this distance is weighted linearly or quadratically. The denominator is used to normalize weights such that the sum of weights in both the sensitive and resistant class equals 0.5. Here, N is the total number of samples and the indicator function $I(d_n = d_i)$ is 1 if the class of sample n (denoted as d_n) is equal to the class of sample i and 0 otherwise.

Effect of sample weights on RF model: Once a sample weighting scheme is chosen (*simple*, *linear*, or *quadratic*), the sample-specific weights for each node v in the random forest can be derived. To this end, let $\delta(v)$ denote the set of bootstrap samples reaching node v. Then, the weight w_i^v of sample x_i in node v is computed as:

$$w_i^v = \frac{w_i^*}{\sum_{n \in \delta(v)} w_n^*} \tag{6.5}$$

The denominator ensures that the sum of weights over all bootstrap samples in node v is equal to 1.

We utilize the calculated sample weights in two ways: (1) to influence the splitting of nodes and (2) to influence the prediction in a leaf node. While the full mathematical details on random forest are provided in Section 4.2.4, we summarize the core ideas and most important equations here.

When splitting a node in a tree, we derive a feature X_j and cutpoint s to assign the samples in node v to a left child node v_l and a right child node v_r. Samples where the value of feature X_j is smaller than s are assigned to v_l. Otherwise, they are assigned to v_r. For regression, the feature and cutpoint are chosen to maximize:

$$w_{an}(v) \cdot \big(MSE(v) - w_{ch}(v_r) \cdot MSE(v_r) - w_{ch}(v_l) \cdot MSE(v_l)\big) \tag{6.6}$$

The factors w_{an} and w_{ch} are node-specific weights for the ancestor (parent) and child nodes, respectively, which we can be defined as follows by using our sample weights:

$$w_{an}(v) = \frac{\sum_{i \in \delta(v)} w_i^*}{\sum_{i \in \delta(root(v))} w_i^*} \tag{6.7}$$

$$w_{ch}(v) = \frac{\sum_{i \in \delta(v)} w_i^*}{\sum_{i \in \delta(parent(v))} w_i^*} \tag{6.8}$$

Here, $root(v)$ denotes the root node of the tree that v belongs to, and $parent(v)$ denotes the parent node of v. Note that $MSE(v)$ also considers the sample-specific weights:

$$MSE(v) = \sum_{i \in \delta(v)} w_i^v \cdot (y_i - \hat{y}^v)^2 \tag{6.9}$$

$$\text{with } \hat{y}^v = \sum_{i \in \delta(v)} w_i^v \cdot y_i \tag{6.10}$$

When making a tree-specific prediction $\hat{f}_b(x)$ for a sample x reaching leaf node μ in tree b, the sample weights are used to obtain a weighted mean:

$$\hat{f}_b(x) = \sum_{i \in \delta(\mu)} w_i^{\mu} \cdot y_i \tag{6.11}$$

As discussed above, we can use our regression random forest to also make classification predictions, since the class $d_i \in \{sensitive, resistant\}$ of each sample i in the tree is known. To this end, a weighted majority vote over all samples in the reached leaf node is computed:

$$\hat{f}_b(x) = \underset{c \in \{\text{sensitive,resistant}\}}{\operatorname{argmax}} \left\{ \sum_{i \in \delta(\mu)} w_i^{\mu} \cdot I(d_i = c) \right\} \tag{6.12}$$

As an alternative to sample-specific weights, upsampling can be performed before training the model, which will be discussed in the following.

Simple Upsampling: The idea behind upsampling is to modify the training data by (repeatedly) duplicating certain samples, thereby increasing their impact on model training. The simplest form of upsampling we can apply to our problem is drawing with replacement from the set of sensitive cell lines (i.e., the minority class) until the number of sensitive and resistant cell lines in the training data is equal.

Proportional upsampling: Instead of the simple upsampling presented above, we can also consider the continuous drug response values and upsample the most sensitive samples more heavily: First, the linear sample weights w_i^* as given by Equation 6.4 with $g = 1$ are computed for each sensitive sample x_i. The number of duplicates of sample x_i is then determined as $2 \cdot w_i^* \cdot N_{\text{Res}}$, where N_{Res} denotes the number of resistant samples. Since the sum of w_i^* over all sensitive samples is 0.5, the factor 2 ensures that the number of sensitive samples after upsampling is equal to the number of resistant samples (which are not upsampled).

6.1.5.3 Tree Weighting Schemes

In HARF, only the trees that predicted the majority class of a sample are used to predict its continuous drug response (cf. Equation 6.2). We call this weighting scheme *binary tree weights*. Below, we propose several alternative weighting schemes we investigated to enhance regression predictions.

Binary sensitive tree weights: For this weighting, we use the same binary tree weights as HARF for samples predicted to be sensitive but employ the tree weights of a conventional random forest (i.e., $\frac{1}{B}$) for samples predicted to be resistant.

Majority tree weights: For this weighting, we consider not only the majority vote of each tree but the (weighted) fraction of samples in the reached leaf node that agree with the class prediction of the entire forest for sample x:

$$\text{frac}_b(x) = \frac{\sum_{n \in \delta(\mu)} I_n(x) \cdot w_n^*}{\sum_{n \in \delta(\mu)} w_n^*} \tag{6.13}$$

Here, μ denotes the reached leaf node in tree b. The indicator function $I_n(x)$ is 1 if the class of sample n is equal to the class prediction of the entire forest for sample x and 0 otherwise. The final tree weights are then computed as:

$$w_b(x) = \frac{\text{frac}_b(x)}{\sum_{\beta=1}^{B} \text{frac}_\beta(x)} \tag{6.14}$$

Majority sensitive tree weights: For this weighting, we employ the weights given in Equation 6.14 for samples predicted to be sensitive, and the tree weights of a conventional random forest (i.e., $\frac{1}{B}$) for samples predicted to be resistant

6.1.6 Implementation

Elastic nets and boosting trees were trained in R using the *glmnet* (v.4.1.1) [327] and *gbm* (v.2.1.8) [329] packages, respectively. Neural networks were trained in Python using *tensorflow* (v.1.13.1) [330] with GPU support and the *keras* API (v.2.3.1) [331]. All random forests, including HARF and SAURON-RF, were also implemented in Python using the *scikit-learn* package (v.1.0.1) [360], specifically the RandomForestClassifier and RandomForestRegressor classes. The hyperparameters for all models are listed in Appendix Table D.1. Our implementations are available at `https://github.com/unisb-bioinf/SAURON-RF`.

6.2 Results

In the following sections, we present the results of applying SAURON-RF to the GDSC data. First, we show how regression imbalance affects different ML algorithms and how SAURON-RF can counteract it. Next, we compare the performance of SAURON-RF to the performance of different types of random forests across all 86 investigated drugs. Finally, we take a closer look at the input features (i.e., genes) that have a high feature importance (i.e., strong impact on model predictions) in our SAURON-RF models and discuss how they are related to known mechanisms of drug sensitivity or resistance.

6.2.1 Regression Imbalance Across ML Algorithms

To investigate the impact of regression imbalance on different ML algorithms, we trained regression models for the prediction of logarithmized IC50 values for 86 drugs from the GDSC using boosting trees, elastic nets, neural networks, and random forests. On average, the performance of all algorithms is similar, with slightly smaller MSEs for elastic nets and random forests (cf. Appendix Figure D.1). While all four algorithms tend to predict IC50s close to the mean IC50 of the training data quite accurately, IC50s that are increasingly smaller/larger than the mean are increasingly over-/underpredicted. An example of this regression imbalance is shown in Figure 6.2A, which depicts the predicted IC50s and prediction errors for the drug 5-Fluorouracil. With only 8.6% of cell lines reacting sensitively to 5-Fluorouracil, this drug is a typical example of class imbalance in the GDSC data, where the average sensitive-to-resistant ratio per drug is around 1:10.

In an attempt to counteract the regression imbalance, we applied two versions of the HARF algorithm: As the original HARF algorithm uses cancer types as class information, we first trained a HARF model using haematopoietic/lymphoid cell lines and lung cell lines as classes. We chose these cancer types as they are the most abundant in the GDSC. As shown in Figure 6.2B, HARF is able to clearly separate the two classes. Predictions for the haematopoietic/lymphoid class (mean ln(IC50) for 5-Fluorouracil: 3.03) are much lower than those for the lung class (mean ln(IC50): 5.45). However, the tendency to over-/underpredict small/large IC50s persists. Furthermore, by only considering these two cancer types, the number of available cell lines is reduced drastically from 806 to 274 (139 haematopoietic/lymphoid, 135 lung).

Next, we applied HARF again but changed the class definition from cancer types to cell lines that are *sensitive* or *resistant* to 5-Fluorouracil. Consequently, the classification and regression problems are closely linked since sensitive cell lines, by definition, have a smaller IC50 than resistant cell lines. Moreover, this setting – which we call HARF$_{\mathrm{SR}}$ – allows us to again consider all 806 available cell lines for model training and testing. However, Figure 6.2B shows that HARF$_{\mathrm{SR}}$ only classifies one sensitive cell line correctly, which is likely caused by the large class imbalance. Again, over-/underprediction occurs, which is even more extreme for HARF$_{\mathrm{SR}}$ than for the original HARF approach. Consequently, changing class definitions without accounting for the class imbalance seems insufficient to improve predictions for the sensitive cell lines.

The right-most plot of Figure 6.2B shows the results of applying SAURON-RF with simple sample weights and binary sensitive tree weights (cf. Section 6.1.5): A much larger number of sensitive cell lines is correctly identified compared to HARF$_{\mathrm{SR}}$ leading to significantly smaller prediction errors for these samples. This improvement comes at the cost of some resistant cell lines being falsely identified as sensitive and, consequently, underpredicted.

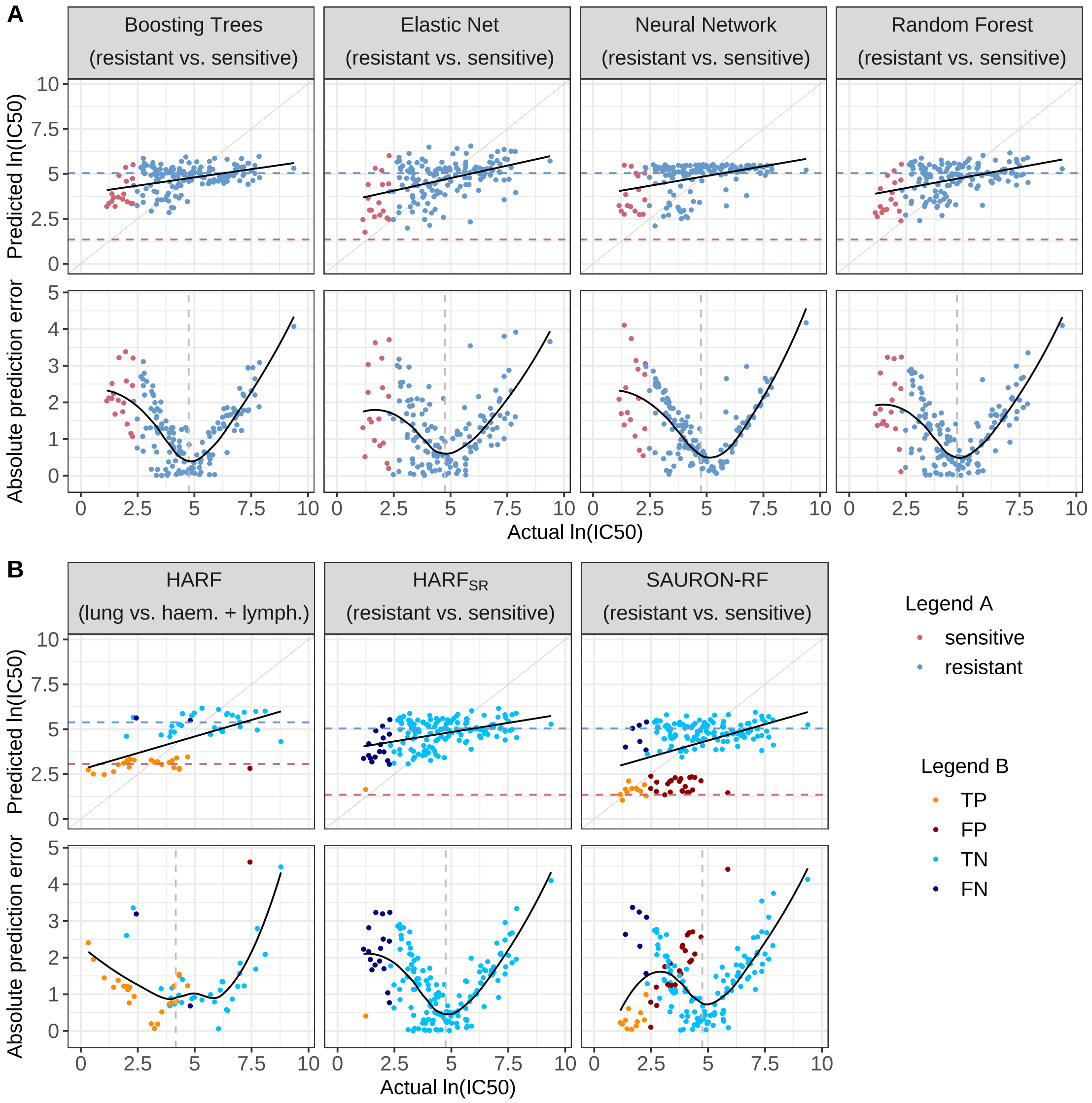


FIGURE 6.2: Regression performance of different ML algorithms for 5-Fluorouracil. Subfigure A depicts results for four conventional ML approaches: boosting trees, elastic net, a neural network, and a random forest. Subfigure B depicts results for HARF, where the cancer types haematopoietic/lymphoid and lung are considered as classes, HARF-SR, where *sensitive* and *resistant* are considered as classes, and SAURON-RF (with simple sample weights and binary sensitive tree weights), which also uses *sensitive* and *resistant* as classes. The upper row of each subfigure depicts the predicted vs. actual ln(IC50) values. The solid black line is a regression line fitted to the data and the dashed lines represent the mean ln(IC50) of all training samples of the considered classes. The lower row of each subfigure depicts the absolute prediction error vs. the actual ln(IC50). Here, the solid black cure is a loess curve fitted to the data, and the gray dashed line represents the mean ln(IC50) of the training data. In Subfigure A, points are colored according to the actual class (*sensitive*/*resistant*) of each cell line. In Subfigure B, points are colored according to the respective model's prediction as true positive (TP), false positive (FP), true negative (TN), and false negative (FN). For the original HARF algorithm, the positive class (P) corresponds to the haematopoietic and lymphoid cell lines and the negative class (N) to the lung cell lines. For HARF$_{SR}$ and SAURON-RF, the positive class corresponds to the sensitive cell lines and the negative class to the resistant ones.

In the following section, we systematically assess the performance of different variations of SAURON-RF across all investigated drugs and compare the results to conventional regression/classification random forests and HARF_{SR}.

6.2.2 Performance Comparison for Regression and Classification

We used different variations of regression/classification random forests, HARF_{SR}, and SAURON-RF to train models for 86 drugs of the GDSC dataset. The results in terms of test set performance averaged across all drugs are shown in Figure 6.3 (additional results are shown in Appendix Figures D.1 to D.3). To measure regression performance, we considered the MSE and median squared error (median SE) across all cell lines of the test set but also considered the MSE and median SE for the groups of sensitive and resistant cell lines separately. To measure classification performance, we used Matthew's correlation coefficient (MCC), sensitivity (fraction of correctly identified sensitive samples), and specificity (fraction of correctly identified resistant samples). To obtain regression predictions from a classification random forest, we average the ln(IC50) values of training samples in the reached leaf node of each tree and then average predictions over all trees. To obtain class predictions from a regression random forest, we first use the forest to predict an ln(IC50) for each cell line, which we then binarize using the drug-specific IC50 thresholds. (cf. Section 6.1.1)

Across all investigated algorithms, regression and classification performance is worse for the group of sensitive cell lines compared to the group of resistant cell lines, as shown in Figure 6.3: The average MSE for sensitive cell lines (*sensitive MSE*) is always considerably larger than for resistant cell lines (*resistant MSE*) and the test sensitivity is always much lower than the test specificity.
For the conventional regression RF (rRF) and HARF_{SR}, the sensitive MSE is 2.9 and 3.2 times larger than the resistant MSE, respectively. Furthermore, the sensitivity for both approaches is only around 10%, while specificity is 99%. Through the addition of simple sample weights (cf. Equation 6.3), the sensitive MSE can be improved by 32.8% (rRF)/26.6% (HARF_{SR}[1]). Likewise, sensitivity is improved strongly by 21%/50%. However, these substantial improvements for sensitive cell lines come at the cost of some performance loss for the resistant cell lines: the resistant MSE is increased by 23.5% (rRF)/29.8% (HARF_{SR}) and specificity is decreased by 5%/15%. Similar but slightly worse results can be observed when using upsampling of sensitive cell lines instead of sample weights.

[1] Note that the HARF_{SR} model with the addition of simple sample weights is denoted as "SAURON-RF, simple sample weights, binary tree weights" in Figure 6.3.

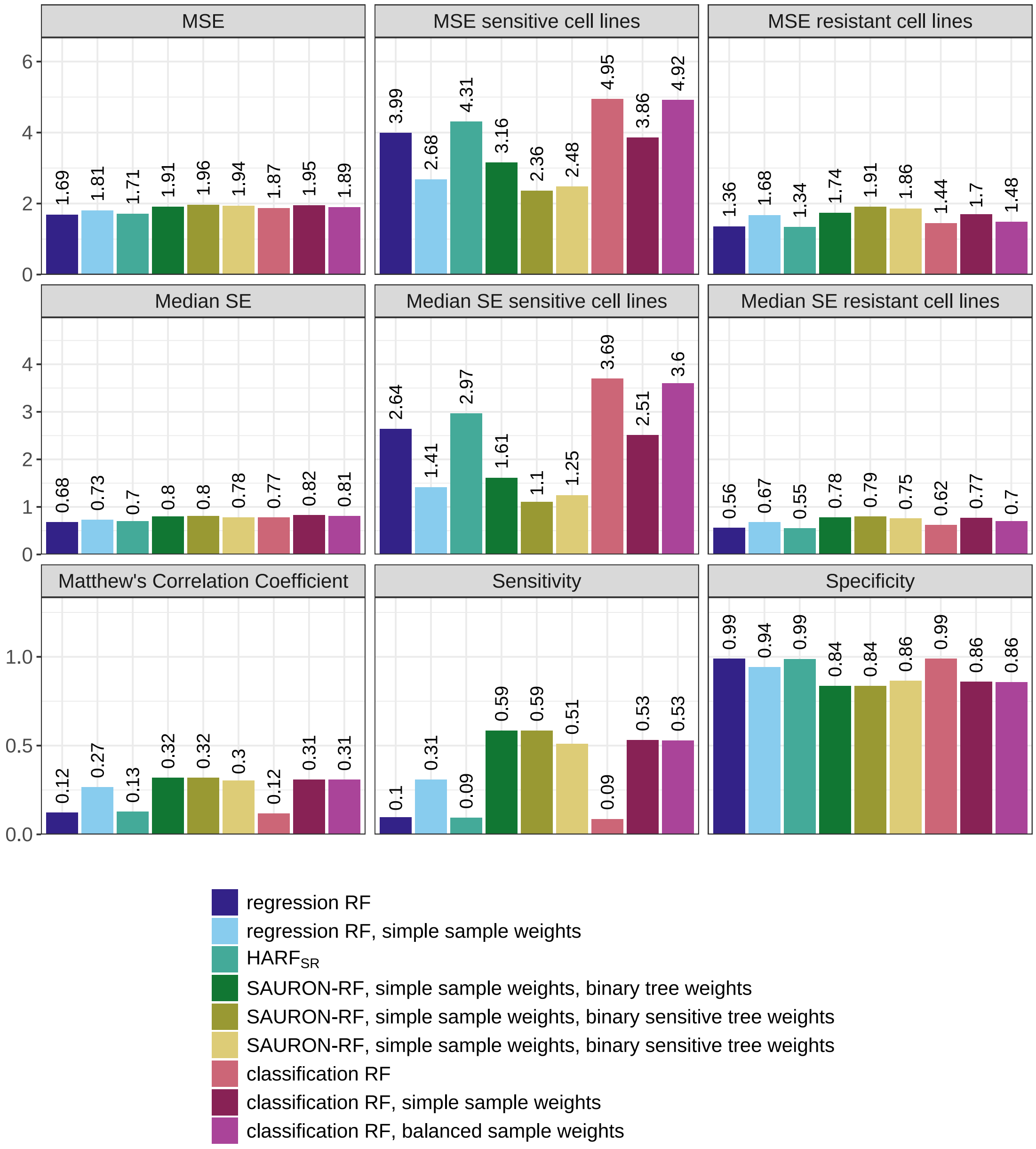


FIGURE 6.3: Regression and classification performance across drugs. In this figure, we compare the regression and classification performance of regression random forests, classification random forests, HARF$_{SR}$ and various versions of SAURON-RF. All error measures are calculated on the drug-specific test sets and averaged across all 86 investigated drugs.

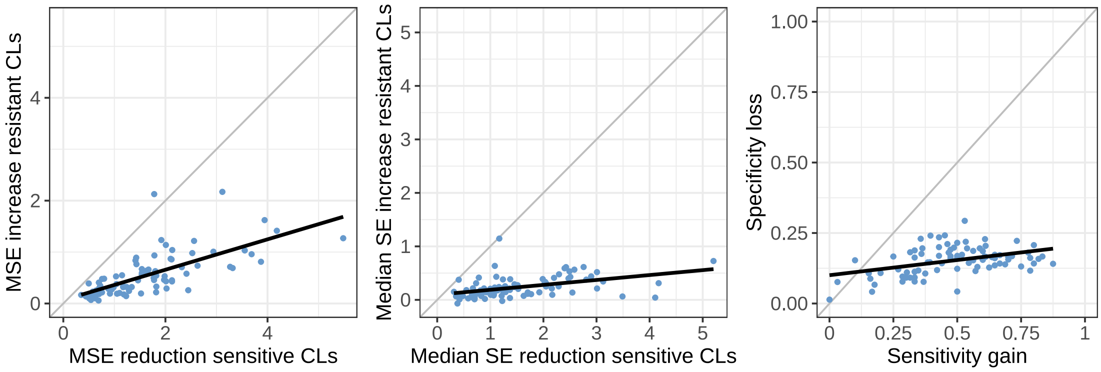


Figure 6.4: Performance trade-off between SAURON-RF and regression random forest. This figure shows the performance trade-off between predictions for sensitive and resistant cell lines (CLs) when using SAURON-RF (simple sample weights, binary sensitive tree weights) compared to a conventional regression random forest based on different regression/classification error measures. Each point in the scatter plots corresponds to one drug. Additionally, a regression line fitted to the data is shown in black.

The overall best performance for regression and classification was obtained for SAURON-RF using simple sample weights and binary sensitive tree weights: Compared to the conventional regression RF, the sensitive MSE is improved by 40.8%, while the resistant MSE is increased by a similar amount (40.4%).

Next, we compared SAURON-RF to conventional classification random forests (cRF). Similar to the conventional regression RF and HARF$_{\mathrm{SR}}$, cRF has a low sensitivity (9%) but high specificity (99%). When adding simple sample weights or the sample weights of the *scikit-learn* random forest package (called *balanced* sample weights in Figure 6.3) to cRF, the sensitivity increases notably (to 53%) but remains lower than the sensitivity for SAURON-RF (59%). Specificity is comparable for both approaches (86% cRF, 84% SAURON-RF). Regarding regression performance, the MSE for sensitive cell lines using the best-performing cRF (cRF with simple sample weights) is 64% larger than for the best-performing version of SAURON-RF.

Lastly, we compared SAURON-RF to a hierarchical RF approach. To this end, we trained one classification RF to predict whether a cell line is sensitive or resistant. Next, we trained two regression RFs on the subsets of sensitive and resistant cell lines, respectively, to predict ln(IC50) values. Depending on the prediction of the classification RF, the corresponding regression RF is used for the continuous prediction. This approach was inferior to all versions of SAURON-RF presented in Figure 6.3. Details are shown in Appendix Figure D.3.

Our results show that SAURON-RF improves both regression and classification performance for the group of sensitive cell lines. However, this improvement comes at the cost of increasing errors for the resistant cell lines. In Figure 6.4, we assess this performance

trade-off for our best-performing version of SAURON-RF compared to a conventional regression RF. For almost all drugs, the improvements for sensitive cell lines when using SAURON-RF outweigh the performance loss for resistant cell lines. This observation holds for both classification and regression measures.
Another noteworthy observation is that the MSE of SAURON-RF for sensitive cell lines that were correctly classified is three times smaller than the MSE of all sensitive cell lines. Thus, if we could improve the classification performance further, this should also entail a strong improvement in regression performance.

Since the analyses presented in this section focused only on RF-based approaches, we would like to highlight that SAURON-RF also outperforms the ML algorithms investigated in the previous section (elastic net, boosting trees, and neural networks) regarding the prediction performance of sensitive cell lines. As shown in Appendix Figure D.1, the average sensitivity across the 86 tested drugs is ≤ 0.18 and the MSE of sensitive cell lines is ≥ 3.65 for all three algorithms.

6.2.3 Investigation of Predictive Biomarkers

A benefit of tree-based models is that the if-then-else structure of individual trees makes them relatively easy to interpret (cf. Figure 4.1). However, the manual inspection of an entire forest comprising hundreds of trees becomes practically infeasible. Thus, methods to aggregate the impact of features for an entire forest through so-called feature importance scores exist [236, 361]. One such method implemented in the *scikit-learn* Python package measures feature importance by quantifying the total error reduction (cf. Equation 6.6) that was obtained from decisions involving a certain feature across all trees in the forest [362]. In the following, we use these feature importance scores to investigate whether features with high importance in our models are linked to known mechanisms of drug sensitivity or resistance. To this end, we considered the ten drugs, for which the MCC using the best-performing version of SAURON-RF was the largest (cf. Appendix Figure D.4). For each drug, we identified the five most important features. We then conducted a literature search to determine whether the identified genes are known to impact treatment response either positively or negatively. For six out of ten drugs, we found at least one such feature among the five most important ones (see Appendix Tables D.2 to D.7). In the following, we exemplarily discuss our findings for the drugs ABT737 and Nutlin-3a(-) in more detail.
ABT737 inhibits the apoptosis-regulator BCL2, and BCL2 was among the five most important features for this drug. Other important features for ABT737 include MIR22HG, which is involved in downregulating BCL2 [363, 364] and IDH2, which was shown to increase ABT737 sensitivity when mutated [365, 366]. Additionally, we identified BLVRB

as important. While we found no direct association of BLVRB to ABT737 directly, BLVRB has been linked to increased sensitivity to Obatoclax, which is also a BCL2 inhibitor [133].

The drug Nutlin-3a(-) targets the p53 pathway, more specifically the MDM2 oncogene [367], which was among the most important features for this drug. Additionally, we identified RPS27L as important, which is a downstream target of TP53 (*Tumor Protein p53*). RPS27L was shown to be active in cells that undergo apoptosis following Nutlin-3a(-) treatment [368]. Lastly, we identified DDB2 and CYFIP2, for which an increase in expression was observed after treatment with Nutlin-3a(-) [369, 370]. Consequently, these genes might be involved in the mode of action of Nutlin-3a(-).

For some drugs, we were unable to identify any previously-reported markers of treatment sensitivity or resistance among the most important features. One explanation may be that such markers are not always known or at least not publicly available. Hence, the most important features for these drugs in our models might hint at currently unknown response mechanisms. It is also possible that some known markers were eliminated by our feature selection, e.g., when highly correlated features are selected instead.

Lastly, we investigated which genes were considered important across multiple drugs. For each gene, we counted for how many drugs it was selected through the MRMR feature selection and computed its average feature importance score. We identified three genes for which both the number of drugs and the average feature importance were relatively high compared to the other genes (cf. Appendix Figure D.5): PPIC, SDC4, and DCBLD2. While PPIC is involved in protein folding [371], the transmembrane proteins SDC4 and DCBLD2 play a role in cellular signaling [372, 373]. The upregulation of DCBLD2 was linked to 5-Fluorouracil resistance [373] and to poor prognosis in various cancers [373–375]. However, the specific mechanisms through which these genes confer or alleviate multi-drug resistance remain to be investigated. Note also that none of these genes was chosen for more than 20% of drugs, so they might only be markers of drug response for certain types of drugs, e.g., drugs with the same target molecule or pathway.

6.3 Discussion

The underrepresentation of sensitive samples in drug response data poses a challenge for ML since it negatively affects both the classification and regression performance of these samples. However, since sensitive samples represent cases of putative treatment success, their accurate prediction is of particular interest for personalized medicine. To address this issue, we developed SAURON-RF, a method that simultaneously performs classification and regression to predict drug responses. SAURON-RF substantially improves predictions for the underrepresented group of sensitive cell lines compared to conventional ML algorithms, including various types of random forests, as well as HARF_{SR}, and a hierarchical prediction approach. However, just using sample weights in combination with conventional random forests already results in relatively strong performance improvements.

Our results also show that the accurate classification of a sample can substantially improve its regression prediction in SAURON-RF. Consequently, designing models that combine classification and regression seems to be a promising direction for future research. It could, for example, be beneficial to simultaneously optimize a classification and a regression criterion during model training.

The improvements of SAURON-RF for sensitive cell lines come at the cost of a performance loss for resistant cell lines. Even though we showed that the improvements for the sensitive samples outweigh the performance decrease for the resistant samples, increasing the errors for resistant samples too much can lead to these samples being falsely predicted as sensitive. Such mistakes should be avoided since, in a real-life scenario, they could lead to recommending an ineffective treatment to a patient. Still, the classification and regression performance of the resistant cell lines remains superior to that of the sensitive cell lines, even for SAURON-RF. In the next chapter, we show how both *false sensitive* and *false resistant* misclassifications can be reduced by applying conformal prediction (cf. Section 4.3.2.1) to SAURON-RF.

Another approach to potentially reduce errors is to consider additional or alternative input features: While we identified the MRMR feature selection as one of the best-performing approaches to select gene expression features in the previous chapter, investigating other data types such as (epi-)genomic features [176], protein-protein interactions [293], pathway activities [49], or pharmacogenomic a priori knowledge [37], might improve predictions further. Besides the cell line characterization in the model input, the utilized drug response measures can also be modified: In the next chapter, we extend the classification problem to three classes (*sensitive*, *intermediate*, *resistant*), which reduces the risk of misclassifications between the sensitive and resistant classes. Additionally, we propose a novel continuous drug sensitivity measure that, unlike the IC50, is comparable across drugs (cf. Section 3.2.4.1). This enables the use of SAURON-RF to perform drug prioritization, a major step toward achieving patient-specific treatment recommendations.

Eventually, we envision SAURON-RF to be integrated into decision support tools like $\text{ClinOmicsTrail}^{\text{bc}}$, which focuses on breast cancer treatment [376]. $\text{ClinOmicsTrail}^{\text{bc}}$ analyzes clinical and molecular patient data (e.g., genetic alterations or changes in pathway activity that may be known to affect drug effectiveness) with the goal of deriving a personalized treatment strategy. Here, SAURON-RF could aid in assessing the suitability of standard-of-care drugs but may also identify promising alternative treatment options.

Chapter 7

Reliable SAURON-RF: Providing Certainty Guarantees for Drug Sensitivity Prediction & Prioritization

In Chapter 4.3, we have discussed which requirements machine learning (ML) models should fulfill to be considered trustworthy. These include good performance, the possibility for humans to interpret predictions, and accounting for data-related challenges such as class imbalances or high dimensionality. Thus, in the two previous chapters, we extensively evaluated the performance of various ML models, and addressed issues such as the underrepresentation of drug-sensitive samples and the selection of interpretable model inputs. A last building block of trustworthiness we did not yet address is *reliability*, i.e., the trust we have in individual predictions for previously unseen samples.

Evaluating model performance on a test set by comparing actual and predicted responses can provide some insight into the performance of a model for previously unseen data. However, we lack a guarantee or estimation of how reliable individual predictions are for unseen samples when no known response is available. Making predictions for individual, previously unseen patients would be the real-life use case for our models. Thus, some form of certainty guarantee or reliability estimation is required to make our ML models trustworthy for this application.

To the best of our knowledge, there exists only one approach that aims to achieve reliable drug sensitivity predictions (cf. Table 7.1): Fang et al. use a quantile regression random forest to predict drug response intervals instead of point predictions [290]. Narrower intervals correspond to cases where the response can be predicted more precisely. Unfortunately, their approach is limited to regression using random forests. Furthermore, it does not provide any guarantee that the actual response of a sample is contained in its predicted interval.

In Section 4.3.2.1, we have discussed conformal prediction (CP) as a means to achieve reliable ML predictions. Briefly recapitulated, instead of point predictions, CP yields prediction sets (classification) or intervals (regression). For an unseen sample, the CP predictions are guaranteed to contain its actual response with a probability of almost exactly $(1 - \alpha)$, where $\alpha \in [0, 1]$ can be chosen freely. This is also known as the CP *certainty guarantee*. Small sets and narrow intervals indicate relatively precise predictions, and set/interval size typically increases for samples where prediction-making is more challenging. CP has already been successfully applied in drug discovery and drug toxicity prediction [272, 377–379].

In this chapter, we pioneer the use of CP for drug sensitivity prediction. We developed a Python CP pipeline that can be used for both classification and regression. It is applicable to arbitrary ML models, given that they provide a notion of prediction uncertainty, e.g., class probabilities for classification. We demonstrate the capabilities of our pipeline by applying it to our joint classification and regression approach SAURON-RF (cf. Chapter 6). Since SARON-RF is based on random forests, class probabilities can directly be obtained as the fraction of trees that voted for a certain class in classification. To estimate regression uncertainties, we extended SAURON-RF with functionality to perform quantile regression, following an algorithm by Meinshausen et al. [380]. Moreover, we extended SAURON-RF to be applicable for multi-class classification.

In addition to introducing CP to the drug sensitivity realm, this chapter addresses another major goal of personalized medicine, namely the challenge of drug prioritization: Drug prioritization describes the task of identifying which drugs are potentially effective to treat a patient and ranking these drugs by their effectiveness. As shown in Table 7.1, most ML approaches in the realm of drug sensitivity research, including our methods presented in Chapters 5 and 6, do not perform any ranking of drugs. Instead, they simply predict which cell lines are sensitive to a certain drug via classification or quantify the degree of sensitivity via regression.

Two approaches that are specifically designed to rank the most effective drugs for a given cell line are KRL by He et al. [288] and PPORank by Liu et al. [215]. However, both approaches suffer from the same drawbacks: They do not predict drug sensitivities directly but only the ranking of drugs, which He et al. call *drug recommendation*. Consequently, differences in effectiveness between the ranked drugs cannot be assessed. Furthermore, both approaches do not attempt to distinguish effective from ineffective treatments before ranking. Instead, they always predict rankings of a fixed length k. Thus, if the number of supposedly effective drugs is smaller than k, the ranking would contain ineffective drugs.

In this chapter, we want to overcome the challenges of existing approaches and finally enable reliable drug *prioritization* by proposing a novel prioritization framework: For a given cell line, we apply SAURON-RF to first identify potentially effective drugs via classification and then rank them via regression. A previously unresolved challenge of ranking drugs for a cell line is that common response measures like the IC50 or AUC are not comparable between drugs (cf. Section 3.2.4.1). Thus, we developed a novel drug response measure that is comparable between drugs. Our measure – the *CMax viability* – quantifies the effect that a drug achieves at its highest clinically recommended dose. Thus, it is designed to measure the maximum effect a drug can realistically produce. Our evaluations on the GDSC (Genomics of Drug Sensitivity in Cancer) database show that the rankings derived from our prioritization framework contain the most effective drug for a given cell line in most cases. Even if this drug is not ranked first, the first-ranked drug is comparable to the actual best option in terms of effectiveness. Additionally, we incorporated CP into the prioritization procedure. Consequently, we can provide certainty guarantees for our rankings, thereby enhancing their trustworthiness. Additionally, CP reduces the amount of ineffective drugs in the prioritized lists. This is highly desirable since it corresponds to avoiding the recommendation of ineffective treatments.

Author Contributions

This chapter is based on the following publication:

Eckhart, L.*, Lenhof, K.*, Rolli, L. M., Volkamer, A., & Lenhof, H. P. (2024). *Reliable anti-cancer drug sensitivity prediction and prioritization.* Scientific Reports, 14(1), 12303.
DOI: 10.1038/s41598-024-62956-6
(* authors contributed equally)

Kerstin Lenhof conceived the idea of applying conformal prediction to the GDSC data, while I developed the novel drug sensitivity measure and implemented its derivation from the GDSC data. Lisa-Marie Rolli implemented the conformal prediction pipeline, and Kerstin Lenhof implemented the extensions of SAURON-RF. Kerstin Lenhof also drafted the publication manuscript. Andrea Volkamer and Hans-Peter Lenhof supervised the study, which was designed by Kerstin Lenhof and myself. All authors analyzed the results and reviewed the manuscript.

Table 7.1: Overview of the application of reliability estimation and drug prioritization in drug sensitivity prediction literature. For 31 publications, this table lists whether reliability estimation and drug prioritization have been performed in the respective approach. The row-coloring corresponds to the type of (supervised) learning that was employed (blue: classification, white: regression, yellow: classification and regression, purple: methods that predict a ranking instead of a continuous or discrete response).

Publication	Sensitivity measures	Reliability estimation	Drug prioritization
LOBICO (2016) [36]	binarized IC50	✗	✗
Stanfield et al. (2017) [177]	binarized IC50	✗	✗
Deep-Resp-Forest (2019) [179]	binarized IC50, binarized activity area	✗	✗
MOLI (2019) [314]	binarized IC50	✗	✗
MERIDA (2021) [37]	binarized IC50	✗	✗
GraphCDR (2021) [321]	binarized IC50	✗	✗
RAMP (2022) [315]	binarized IC50	✗*	✗
SAURON-RF (2022) [39]	(binarized) IC50	✗	✗
Reliable SAURON-RF (2024) [39]	(binarized) IC50, (binarized) CMax viab.	✓	✓
Menden et al. (2013) [282]	IC50	✗	✗
Zhang et al. (2015) [283]	IC50, activity area	✗	✗
SRMF (2017) [284]	IC50, activity area	✗	✗
HARF (2017) [285]	AUC	(✓)†	✗
HNMDRP (2018) [307]	IC50	✗	✗
Matlock et al. (2018) [286]	AUC	✗	✗
RWEN (2018) [38]	AUC	✗	✗
CDRscan (2018) [289]	IC50	✗	✗
QRF (2018) [290]	activity area	✓	✗
NCFGER (2018) [325]	IC50	✗	✗
DeepDR (2019) [292]	IC50	✗	✗
NetBiTE (2019) [293]	IC50	✗	✗
Deng et al. (2020) [294]	normalized AUC	✗	✗
ADRML (2020) [296]	IC50	✗	✗
PathDSP (2021) [297]	IC50	✗	✗
MinDrug (2021) [381]	IC50	✗	✗
GraphDRP (2022) [299]	IC50	✗	✗
NeRD (2022) [322]	IC50	✗	✗
GADRP (2023) [323]	IC50	✗	✗
mVAEN (2023) [324]	IC50, activity area	✗	✗
KRL (2018) [288]	normalized IC50	✗	(✓)‡
PPORank (2022) [215]	normalized IC50	✗	(✓)‡

* Lee et al. mention the possibility of assessing the prediction certainty of neural networks by using Monte Carlo dropout [382].

† Rahman et al. use the Jackknife-after-bootstrap approach [383] to compute confidence intervals. However, they did not use the approach to assess the reliability of individual predictions.

‡ He et al. and Liu et al. perform drug recommendation, which is related to drug prioritization but not equivalent (see beginning of this chapter). In particular, their approaches do not ensure that the predicted lists contain only effective drugs.

7.1 Materials and Methods

In the following, we first describe the dataset we employed to perform the analyses in this chapter. Next, we present our novel drug sensitivity measure called *CMax viability* and explain how it can be computed from the analyzed data. Afterward, we discuss the in- and outputs of the used ML models as well as the training and testing procedure. Finally, we detail how we extended SAURON-RF to enable performing (1) a multi-class classification and (2) quantile regression, which can be used to predict regression intervals using CP.

7.1.1 Dataset

Data for our analyses was obtained from the GDSC database (Release 8.3), which we described in detail in Section 3.4. More specifically, we downloaded normalized expression values for 17,419 genes and drug screening data in the form of logarithmized IC50 values and raw viability measures for all 367 drugs in the GDSC1 and all 198 drugs in the GDSC2 dataset.

To obtain discrete drug responses from the continuous IC50 values, we binarized them by comparison to a drug-specific threshold t. The thresholds were derived using a method by Knijnenburg et al. [36] described in Section 3.2.4.2. Cell lines with $ln(IC50) \leq t$ are considered *sensitive*, otherwise they are considered *resistant*.

Additionally, we used the raw viability data provided in the GDSC to compute our novel drug sensitivity measure called CMax viability. In the following, we define our measure and detail how it can be calculated.

7.1.2 Our Novel Sensitivity Measure: the CMax Viability

Conventional measures of drug response such as the IC50 or AUC allow comparing the effect of the same drug on different cell lines but not the effect of different drugs on the same cell line (cf. Section 3.2.4.1). Consequently, these measures do not allow prioritizing drugs for one cell line in a straightforward manner. To overcome this problem, we propose a novel drug sensitivity measure with across-drug comparability called *CMax viability*, which is visualized in Figure 7.1. The CMax viability is defined as the relative viability of cell lines after treatment with the CMax concentration of a drug. The CMax concentration is to the peak plasma concentration measured after administering a drug's highest clinically recommended dose [174]. Thus, our measure is designed to estimate the maximal effect a treatment can achieve using clinically feasible doses.

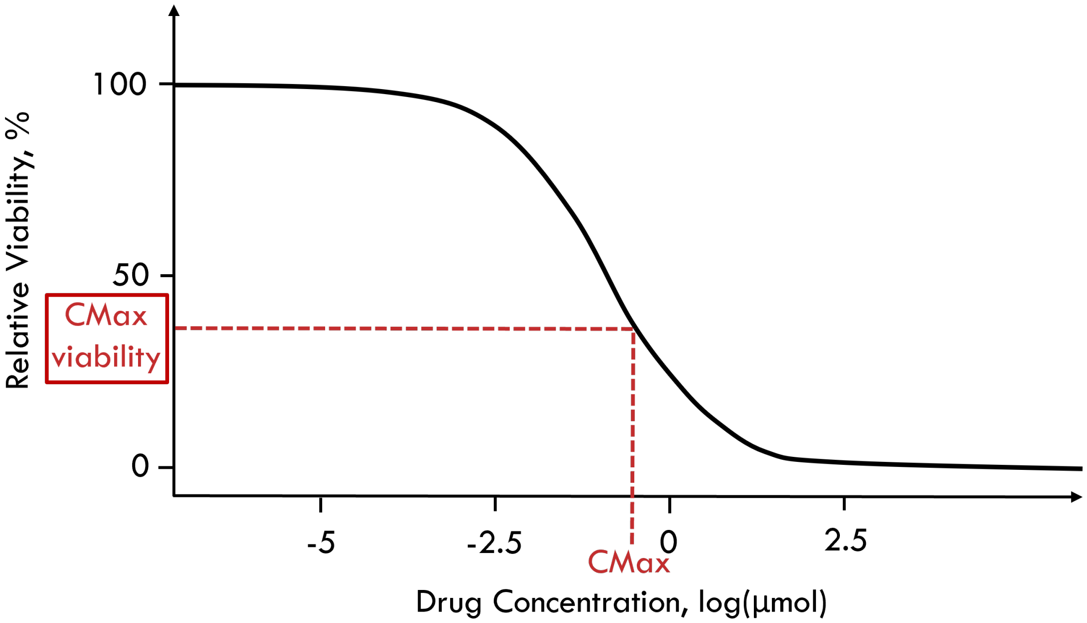


Figure 7.1: CMax viability. This figure shows a dose-response curve, where the x-axis denotes the drug concentration and the y-axis denotes the relative viability. On the x-axis, the CMax concentration of the used drug is marked. The CMax viability is defined as the viability that is reached at the CMax concentration.

As a measure of viability, the CMax viability generally ranges in $[0, 1]$ with smaller values indicating a higher sensitivity (cf. Section 3.2.2). Since our measure is not only comparable across cell lines but also across drugs, it finally enables drug prioritization, which will be investigated in Section 7.2.2.

To compute CMax viabilities for the GDSC data, we obtained CMax concentrations for 60 of the GDSC1 and 47 of the GDSC2 drugs from Liston and Davis [174]. Next, we computed dose-response curves for each cell line-drug combination using a multilevel mixed effects model by Vis et al. [166]. This model was also used to derive the curves for the computation of IC50 values provided in the GDSC (cf. Section 3.4). Briefly summarized, Vis et al. model dose-response curves as two-parametric logistic functions (cf. Equation 3.4 with $\alpha_l = 1$ and $\alpha_h = 0$), where one parameter corresponds to the curve's slope and the other to the inflection point. The slope parameter is constant across all curves for one cell line. Consequently, to determine a cell line's slope parameter, its dose-response data across all available drugs are taken into account.

We discarded all cell line-drug pairs for which less than five dose-response measurements were provided. Furthermore, in line with Vis et al. [166], we discarded all curves where the root mean squared error (RMSE) between the actual dose-response measures and the corresponding points on the fitted curve was greater than 0.3.

Finally, given a dose-response function $f(x)$ that returns the relative viability resulting from a drug treatment with concentration x (cf. Section 3.2.3) and the CMax concentration of the respective drug, the CMax viability is simply given by $f(\text{CMax})$. As shown in Figure 7.1, the CMax viability graphically corresponds to the viability at that point where the dose-response curve intersects a vertical line that passes through the CMax concentration. Exemplary dose-response curves for six cell line-drug combinations from the GDSC with the corresponding CMax viability are provided in Appendix Figure E.1. We discretized the computed CMax viabilities into two classes (*sensitive*, *resistant*) and three classes (*sensitive*, *intermediate*, *resistant*) using partitioning around medoids (PAM) clustering as described in Section 3.2.4.2. This method was previously used by Costello et al. to discretize GI50 values for the DREAM Drug Sensitivity Prediction Challenge [176]. We applied the PAM clustering to all available CMax viabilities across drugs to generate two and three clusters, respectively. The midpoints between the cluster medoids are used as classification thresholds. For two classes, we obtained one threshold $t = 0.5$ and classify samples with CMax viability $\leq t$ as *sensitive* and all others as *resistant*. For three classes, we obtained two thresholds $t_1 = 0.32$ and $t_2 = 0.77$. The smaller threshold is used to distinguish between *sensitive* and *intermediate* samples, while the larger threshold is used to identify *resistant* samples:

$$response = \begin{cases} \text{sensitive}, & \text{if CMax viability} \leq t_1 \\ \text{intermediate}, & \text{if } t_1 < \text{CMax viability} \leq t_2 \\ \text{resistant}, & \text{else} \end{cases} \tag{7.1}$$

Note that this procedure results in viability thresholds that are applicable across all drugs, whereas drug-specific thresholds are required to binarize IC50 values, as discussed in Section 7.1.1 above. Visualizations of the CMax viability distributions and the corresponding thresholds are provided in Appendix Figure E.2.

7.1.3 Model In- and Outputs

We used our SAURON-RF algorithm detailed in Chapter 6 to train drug-specific models that simultaneously predict the continuous and discrete drug response of a cell line based on its gene expression. As continuous measures of drug response, we consider either IC50 values or CMax viabilities. As discrete drug responses, we consider the discretization of these measures into two classes (*sensitive*, *resistant*) or three classes (*sensitive*, *intermediate*, *resistant*) as described above.

To derive model inputs, we applied the same feature selection that was already used for SAURON-RF in Chapter 6, which is based on the minimum-redundancy-maximum-relevance principle and described in Section 5.1.4. Using this method, we selected 50 input features (i.e., genes) for each drug.

7.1.3.1 Conformal Prediction

Additionally, we applied conformal prediction (CP) to our models. An overview of the CP pipeline is provided in Figure 7.2 and a detailed mathematical description of CP can be found in Section 4.3.2.1. Briefly recapitulated, instead of point predictions, conformal prediction yields sets of classes for classification and value intervals for regression. These sets or intervals are guaranteed to contain the actual response of a previously unseen sample with a user-defined probability of almost exactly $1 - \alpha$ with $\alpha \in [0, 1]$. Here, we chose $\alpha = 0.1$.

To generate conformal predictions with an ML model, the models needs to provide some notion of prediction (un-)certainty, e.g., class probabilities for classification. Additionally, a calibration dataset is required, which is disjoint from the training and test data of the model. Intuitively, the calibration data is used to estimate a model's prediction uncertainty for unseen samples. Mathematically, we use our model to make point predictions for the calibration data. Next, a score function is applied to these predictions to obtain a distribution of prediction uncertainty. From this distribution, we can obtain a threshold $\hat{q}$ as a modified $(1 - \alpha)$-quantile. This threshold is then used to compute prediction sets or intervals for the samples in the test set. Since $\hat{q}$ depends on α, the choice of α influences the size of the predicted sets or intervals.

To perform CP for classification, we used three CP scores, namely the True Class, Summation, and Mondrian score, which are described in Section 4.3.2.1. All three scores use class probabilities to measure prediction (un-)certainty. SAURON-RF predicts class probabilities as the fraction of trees that voted for a certain class. A higher probability indicates a larger certainty. Thus, the uncertainty associated with a sample's classification is given by the complement of the predicted probability, i.e.:

$$1 - \frac{\#\text{trees that voted for the predicted class}}{\#\text{trees}} \tag{7.2}$$

For regression, we applied a scoring function that measures uncertainty using quantile regression, which is also described in Section 4.3.2.1. Since the original SAURON-RF approach presented in Chapter 6 cannot perform quantile regression, we extended it as discussed in Section 7.1.5 below.

7.1.4 Model Training and Testing

In this chapter, we evaluate our models from two different perspectives: drug-centric and cell line-centric. The drug-centric perspective is similar to our analyses in the previous two chapters: we train drug-specific models and asses their performance on a per-drug basis. However, one could also be interested in assessing the prediction performance for specific cell lines instead of drugs. In particular, we want to measure how accurately we can identify the most effective drugs for a given cell line and rank them by their effectiveness. Such a cell line-centric evaluation was hampered in previous chapters by the fact that the predicted IC50 values are not comparable between drugs. However, since our novel response measure, the CMax viability, is comparable across drugs, a cell line-centric analysis is now possible. In the following, we detail the model training and evaluation process for both the drug-centric and cell line-centric analyses.

7.1.4.1 Drug-Centric Analyses

To ensure a fair comparison between models that use the IC50 values as drug response and models that use the CMax viability, we only considered those drugs for which both measures are available. This results in a total of 60 drugs from GDSC1 and 47 drugs from GDSC2. Furthermore, we excluded all drugs from the analyses, where class imbalances in the discrete drug response were too strong (see Appendix E.1 for details). Overall, we were able to analyze 41 GDSC1 and 32 GDSC2 drugs using the binarized drug responses. For the ternary responses, we were able to analyze 37 GDSC1 and 28 GDSC2 drugs. All drugs are listed in Appendix Tables E.1 and E.2.

To train our drug-specific models, we divide the available cell lines for each drug into a training set (70% of cell lines), a calibration set (15% of cell lines) and a test set (15% of cell lines).

For each drug, one SAURON-RF model is trained on the training data using the hyperparameters provided in Appendix Table D.1. While we presented different versions of SAURON-RF in Chapter 6, we only consider the best-performing version here, namely SAURON-RF with simple sample weights and binary sensitive tree weights. Next, since our models predict a discrete and a continuous drug response, we apply CP to both the classification and regression tasks using the calibration data. As mentioned above, we test three different CP scores for classification and one score for regression. Finally, we use the trained and calibrated models to generate predictions for the test set without CP (i.e., point predictions) and with CP (i.e., sets/intervals).

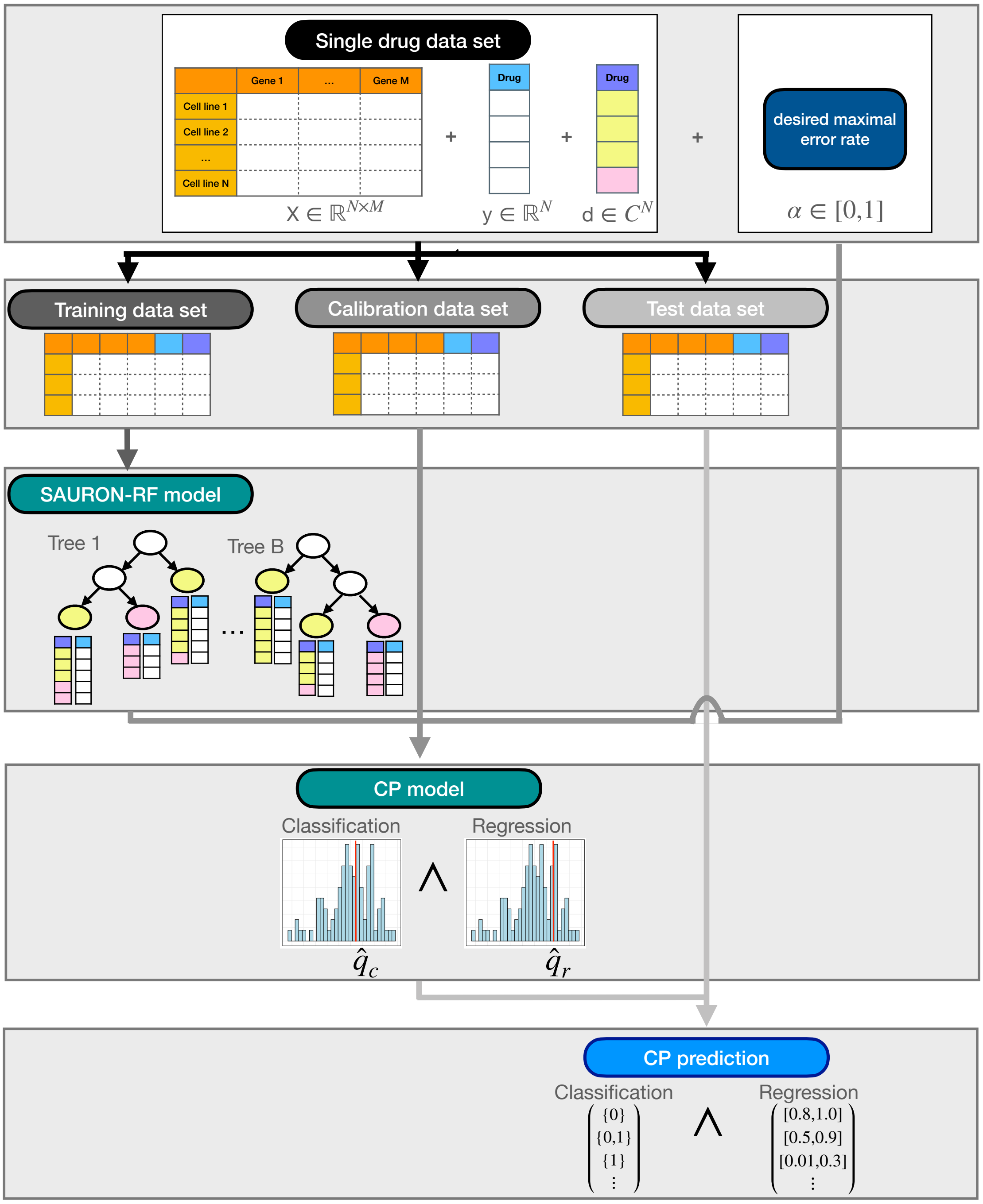


FIGURE 7.2: Conformal prediction pipeline. This figure shows the steps of applying CP to a drug-specific SAURON-RF model. At the top, the available data for one drug is shown, which consists of a feature matrix X (e.g., gene expression values), a continuous drug response y (e.g., the CMax viability), and a discrete drug response d (e.g., classes *sensitive* (1) and *resistant* (0)). The data is first split into a training, calibration, and test set. On the training set, a SAURON-RF model is trained. Next, the calibration data is used to derive distributions of prediction (un-)certainty for regression and classification. Using the desired maximal error rate α, thresholds $\hat{q}_c$ (classification) and $\hat{q}_r$ (regression) are obtained, which can then be used to compute conformal predictions for the test data (cf. Section 4.3.2.1). These prediction sets/intervals are guaranteed to contain the true response of a sample with a probability of almost exactly $1 - \alpha$. This pipeline can not only be applied to SAURON-RF but to any classification/regression algorithm that provides a measure of prediction (un-)certainty.

7.1.4.2 Cell Line-Centric Analyses

For the cell line-centric analyses, we want to investigate the effect of different drugs on the same cell line. In particular, we want to prioritize drugs for a given cell line. These analyses can only be performed using the CMax viabilities since IC50 values are not comparable across drugs. For these analyses, we removed drugs with high class imbalance by only considering drugs with at least 6% of cell lines in each class. Overall, we considered 25 drugs in GDSC1 and 25 drugs in GDSC2, which are listed in Appendix Tables E.1 and E.2, respectively.

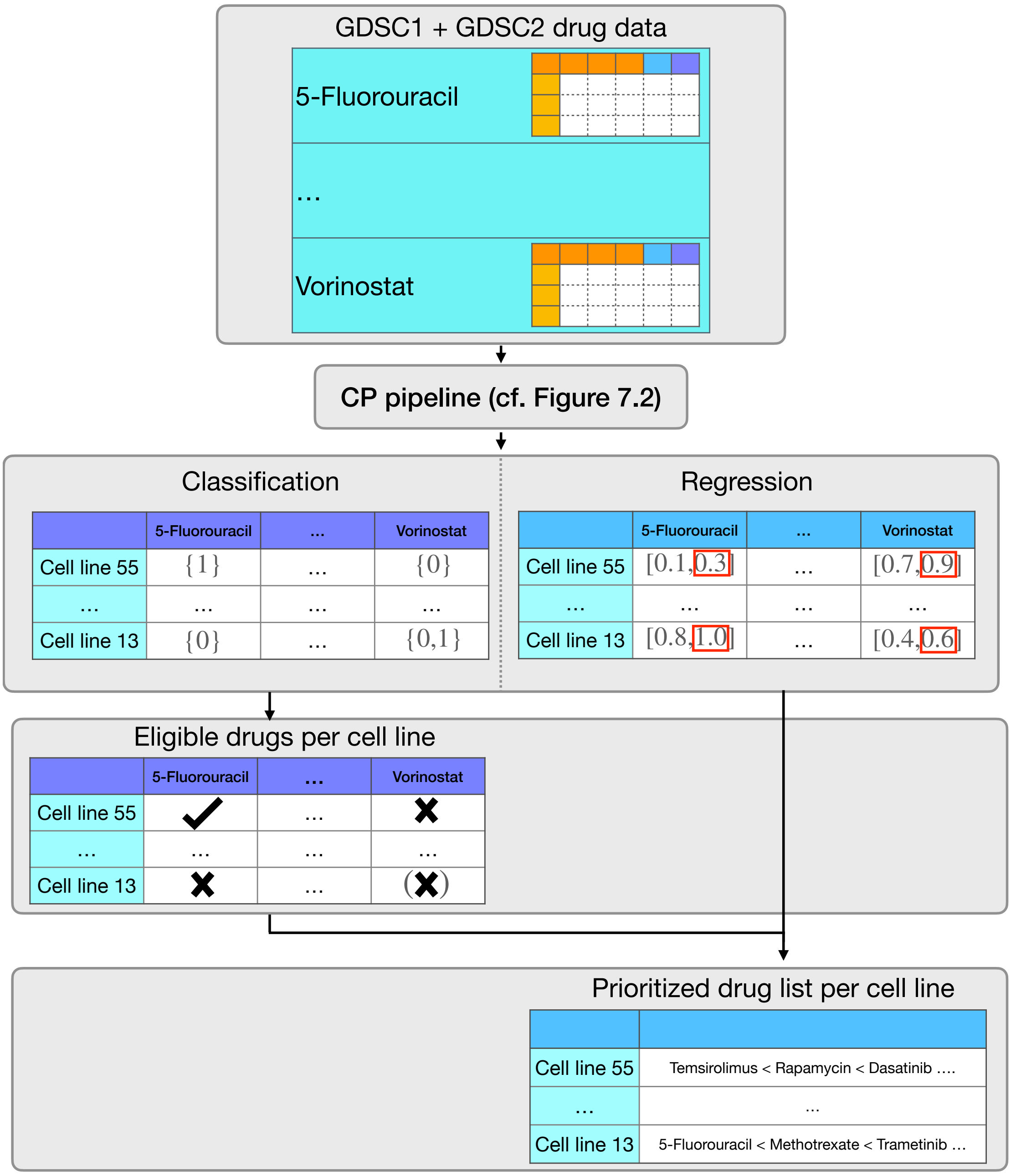
If we want to prioritize drugs for a specific cell line, this cell line should not be used for the training of any of the drug-specific models. Consequently, we selected a test set (and calibration set) of cell lines that is equal for all models. To this end, we identified those cell lines with available drug response for all investigated drugs in the GDSC1 (243 cell lines) and GDSC2 (609 cell lines), respectively. From these cell lines, we randomly sampled a test set and calibration set that are similar in size to the sets for the drug-centric setting (GDSC1: 121 cell lines, GDSC2: 152). For each drug, all cell lines with available response data that are not contained in the calibration and test set are used for training.

The prioritization process with CP is visualized in Figure 7.3 and carried out as follows: For each cell line in the test set, we make predictions using all drug-specific models. Next, we identify the effective drugs for each cell line, i.e., drugs where CP returns a single-class set containing only the sensitive class. Finally, these eligible drugs are sorted ascendingly according to the upper limit of the predicted regression interval.

Note that prioritization can also be performed without CP: we consider all drugs as effective where the classification response is sensitive and rank them according to the point predictions derived from regression.

7.1.5 Extension of SAURON-RF

For the analyses presented in this chapter, we extended our method SAURON-RF in two ways: First, we extended the definitions of sample weights to enable predictions for more than two classes. We will use this to study a three-class separation into *sensitive*, *intermediate*, and *resistant* cell lines. Second, we extended SAURON-RF with the functionality to perform quantile regression [380], which will be used to perform conformal prediction using the regression score presented in Section 4.3.2.1. As discussed above, SAURON-RF does not need to be extended to perform conformal prediction for classification: It provides class probabilities that can directly be employed as a measure of classification certainty.


GDSC1 + GDSC2 drug data
5-Fluorouracil
...
Vorinostat
CP pipeline (cf. Figure 7.2)
Classification
Regression
Eligible drugs per cell line
Prioritized drug list per cell line

	5-Fluorouracil	...	Vorinostat
Cell line 55	{1}	...	{0}
...	...	...	...
Cell line 13	{0}	...	{0,1}

	5-Fluorouracil	...	Vorinostat
Cell line 55	[0.1,0.3]	...	[0.7,0.9]
...	...	...	...
Cell line 13	[0.8,1.0]	...	[0.4,0.6]

	5-Fluorouracil	...	Vorinostat
Cell line 55	✓	...	✗
...	...	...	...
Cell line 13	✗	...	(✗)

Cell line 55	Temsirolimus < Rapamycin < Dasatinib ....
...	...
Cell line 13	5-Fluorouracil < Methotrexate < Trametinib ...

FIGURE 7.3: Drug prioritization pipeline with CP. This figure shows how to obtain cell line-specific prioritized drug lists using SAURON-RF with CP and the CMax viability. First, the CP pipeline as shown in Figure 7.2 is applied to obtain conformal classification/regression predictions for each drug and cell line. Next, for each cell line, we identify those drugs for which the CP interval contains only the sensitive class (here denoted as $\{1\}$), i.e., those drugs for which the model is sufficiently certain that the cell line is sensitive. Next, we rank these drugs using the upper interval of the regression predictions. Here, a lower value corresponds to a more effective treatment. This results in ordered lists of effective drugs for each cell line.

7.1.5.1 Multi-Class Extension

In the original SAURON-RF approach, as presented in Chapter 6, we only considered the binary classification task of identifying cell lines as either *sensitive* or *resistant* towards a certain drug. However, SAURON-RF can be extended to any number of classes, which only requires generalizing the definitions of the sample weights used to counteract class imbalances.

Let $C = \{c_1, \ldots, c_K\}$ be the set containing all K considered classes and let N_{c_j} be the number of samples belonging to class c_j with $j \in \{1, ..., K\}$. W.l.o.g., let c_K be the class containing the relative majority of samples, i.e., $\operatorname{argmax}_{c_j \in C} N_{c_j} = c_K$. Based on these definitions, the simple sample weights (cf. Equation 6.3) for a sample x_i belonging to class c_j can be written as:

$$w_i^* = \frac{N_{c_K}}{N_{c_j}} \tag{7.3}$$

In a similar fashion, the *linear* and *quadratic* weight functions introduced for SAURON-RF in Chapter 6 (cf. Equations 6.4) can be adapted to the multi-class setting. Since we do not apply these weights in the current chapter, we omit their definition here but provide them in Appendix E.2.

7.1.5.2 Quantile Regression with SAURON-RF

As discussed in Section 4.1.1, conventional regression methods typically train model parameters to minimize the sum of squared residuals between the actual and predicted responses. This so-called least squares minimization thus aims to estimate the conditional mean of the response variable Y, i.e., $E[Y|X = x]$ where X denotes a P-dimensional predictor variable (cf. Equation 4.4). In conventional regression random forests, e.g., a prediction is made as the mean over all tree-specific predictions, and the tree-specific predictions themselves are computed as mean over the samples in the reached leaf.

However, one may not only be interested in the conditional mean but the entire conditional distribution $F(y|X = x)$:

$$F[y|X = x] = Pr(Y \leq y|X = x) \tag{7.4}$$

Intuitively, F is the probability that the response for a given x does not exceed y. Knowing F can be useful to, e.g., determine a drug response that is unlikely to be surpassed. Here, we want to use F to determine the dispersion of response values to assess prediction reliability. In the following, we discuss how F can be estimated using quantile regression.

In contrast to least squares regression, quantile regression aims to estimate certain quantiles of the response. The α-quantile with $\alpha \in [0, 1]$ is defined as the smallest response value y for which F is at least α:

$$Q_\alpha(x) = \inf\{y : F[y|X = x] \geq \alpha\} \tag{7.5}$$

To predict quantiles, we use a modified version of SAURON-RF, which is described in the following. Our approach is based on a publication of Meinshausen et al. [380], which discusses how quantile regression can be performed using conventional random forests by estimating F. Let $\mathbf{y} \in \mathbb{R}^N$ denote the vector of continuous response values for N samples. According to Meinshausen (MH), the prediction of a random forest for a sample x can then be described as:

$$\hat{f}_{\text{MH}}(x) = E[Y|X = x] = \sum_{i=1}^{N} w_i(x) \cdot y_i \tag{7.6}$$

Here, y_i is the response of sample i and $w_i(x)$ is a forest-wide sample weight, which we will define below. Thus, Meinshausen et al. model random forest predictions as a weighted mean over the responses of all N training samples. This intuition differs slightly from the conventional notion that a random forest prediction is a (weighted) mean over all B trees in the forest. We also followed this notion in SAURON-RF (SR), where the response of x is predicted as:

$$\hat{f}_{\text{SR}}(x) = \sum_{b=1}^{B} w_b(x) \cdot \hat{f}_b(x) \tag{7.7}$$

Here, $\hat{f}_b(x)$ is the tree-specific prediction and $w_b(x)$ is a tree-specific weight (cf. Section 6.1.5.3). We can, however, proof that both representations are equal, i.e., that $\hat{f}_{\text{SR}}(x) = \hat{f}_{\text{MH}}(x)$. Before we show the stepwise proof, we provide some relevant notations:

- $\delta(\mu_b)$ denotes the set of bootstrap samples reaching leaf node μ in tree b.
- $I_{\delta(\mu_b)}(i)$ is an indicator function that is 1 if sample i reaches leaf node μ_b and 0 otherwise.
- $w_n^{\mu_b}$ denotes the weight of bootstrap sample n in leaf node μ_b as introduced in Equation 4.40 for conventional random forests and in Equation 6.5 for SAURON-RF
- $w_{i*}^{\mu_b}$ refers to the weight of a sample i from the original training data (as opposed to a bootstrapped sample, cf. Section 4.2.4.2) in node μ_b. This weight is calculated

by summing the weights of all bootstrap samples that are equal to sample i:

$$w_{i*}^{\mu_b} = \sum_{i' \in \delta(\mu_b):i'=i} w_{i'}^{\mu_b}. \tag{7.8}$$

Based on these definitions, we show the equivalence of $\hat{f}_{\text{SR}}(x)$ and $\hat{f}_{\text{MH}}(x)$:

$$\begin{aligned}
\hat{f}_{\text{SR}}(x) &= \sum_{b=1}^{B} w_b(x) \cdot \hat{f}_b(x) && \text{Definition of } \hat{f}_{\text{SR}}(x) \text{ in Equation 7.7} \\
&= \sum_{b=1}^{B} w_b(x) \cdot \sum_{n \in \delta(\mu_b)} w_n^{\mu_b} \cdot y_n && \text{Definition of } \hat{f}_b(x) \text{ in Equation 6.11} \\
&= \sum_{b=1}^{B} w_b(x) \cdot \sum_{i=1}^{N} I_{\delta(\mu_b)}(i) \cdot w_{i*}^{\mu_b} \cdot y_i && \text{Definitions of } I_{\delta(\mu_b)} \text{ and } w_{i*}^{\mu_b} \text{ above} \\
&= \sum_{b=1}^{B} \sum_{i=1}^{N} w_b(x) \cdot I_{\delta(\mu_b)}(i) \cdot w_{i*}^{\mu_b} \cdot y_i \\
&= \sum_{i=1}^{N} \sum_{b=1}^{B} w_b(x) \cdot I_{\delta(\mu_b)}(i) \cdot w_{i*}^{\mu_b} \cdot y_i \\
&= \sum_{i=1}^{N} w_i(x) \cdot y_i && \text{Define } w_i(x) = \sum_{b=1}^{B} w_b(x) \cdot I_{\delta(\mu_b)}(i) \cdot w_{i*}^{\mu_b} \\
&= \hat{f}_{\text{MH}}(x) && \text{Definition of } \hat{f}_{\text{MH}}(x) \text{ in Equation 7.6}
\end{aligned}$$

Given this equivalence, we can estimate the conditional distribution function F for SAURON-RF exactly as proposed by Meinshausen [380]:

$$\hat{F}[y|X = x] = \sum_{i=1}^{N} w_i(x) \cdot I_{y_i \leq y} \tag{7.9}$$

The indicator function $I_{y_i \leq y}$ is 1 if $y_i \leq y$ and 0 otherwise. To apply Equation 7.9 for the estimation of F, we need to determine the forest-wide sample weights $w_i(x)$ using the following procedure:

1. Train a SAURON-RF model as described in Section 6.1.5.

2. For a new sample x, determine its reached leaf node μ_b for each tree $b \in \{1, ..., B\}$.

3. For each leaf μ_b, calculate the node-specific weight $w_{i*}^{\mu_b}$ for each training sample x_i with $i \in \{1, ..., N\}$ (cf. Equation 7.8).

4. Compute the forest-wide weight $w_i(x)$ of each training sample x_i by aggregating its node-specific weights obtained in the previous step over all leaf nodes μ_b:

$$w_i(x) = \sum_{b=1}^{B} w_b(x) \cdot I_{\delta(\mu_b)}(i) \cdot w_{i*}^{\mu_b} \quad \forall i \in \{1, \ldots, N\} \tag{7.10}$$

   Only trees where the new sample x reaches the same leaf node as the training sample x_i contribute to $w_i(x)$. If a training sample never reaches the same leaf node as sample x, its weight is 0.

Once the weights $w_i(x)$ are determined, they can be plugged into Equation 7.9 to obtain an estimate $\hat{F}[y|X = x]$ of the conditional distribution function. Next, by plugging $\hat{F}[y|X = x]$ into the definition of the α-quantile in Equation 7.5, we can calculate an estimate of $\hat{Q}_\alpha(x)$ for any α.

By extending SAURON-RF with the functionality to perform quantile regression, we can now perform CP with SAURON-RF using the regression score presented in Section 4.3.2.1.

7.1.6 Implementation

Just like the original SAURON-RF approach discussed in Chapter 6, the extensions discussed here were also implemented in Python using the *scikit-learn* package (v.1.0.1) [360]. The hyperparameters of SAURON-RF are listed in Appendix Table D.1. The conformal prediction pipeline was also implemented in Python, while the CMax viabilities were computed in R. Specifically, we used the *gdscIC50* package (v.0.99.4) [165], which provides an implementation of the algorithm by Vis et al. for the fitting of dose-response curves [166]. Our implementations are available at `https://github.com/unisb-bioinf/Conformal-Drug-Sensitivity-Prediction`.

7.2 Results

In the following sections, we present the results of our drug-centric and cell line-centric analyses for the GDSC2 database. The corresponding results for the GDSC1 can be found in Appendix E.6. For the drug-centric analyses, we first investigate how accurately SAURON-RF with and without CP predicts IC50 and CMax viability values and discretized drug responses derived from these measures. To investigate the capabilities of our CP pipeline for a state-of-the-art method other than SAURON-RF, we apply it to a modified version of the deep neural network by Chiu et al. [292], which we already analyzed in our benchmarking in Chapter 5.

For the cell line-centric analyses, we perform drug prioritization, where we reliably identify effective drugs for a given cell line and then rank these drugs by their predicted effectiveness.

7.2.1 Drug-Centric Analysis – Drug Sensitivity Prediction

For the drug-centric analyses, we first focus on a two-class classification using the IC50 values and CMax viabilities and then discuss a more fine-grained three-class setting. In the two-class setting, the *positive* class (also denoted as class 1) represents the sensitive cell lines, and the *negative* class (0) represents the resistant ones.

7.2.1.1 Two Classes

IC50 values: In our first analysis, we applied SAURON-RF to the IC50 data. In the following, we start by discussing the classification performance and afterward discuss the regression performance with and without CP, respectively.

Figure 7.4 shows the classification performance of SAURON-RF. As expected, the performance of SAURON-RF without CP is comparable to its performance discussed in Chapter 6: The average sensitivity is 56%, specificity is 87%, and Matthew's correlation coefficient (MCC) is 0.35.

Next, we investigated the performance of SAURON-RF with CP using an error rate of $\alpha = 10\%$ in terms of CP *coverage*. Coverage measures the fraction of test samples for which the true class is contained in the predicted class set. The average coverage across drugs is around $1 - \alpha = 90\%$ for all three classification scores. Thus, the CP certainty guarantee is fulfilled. However, when investigating the coverage for the groups of sensitive and resistant samples separately, the certainty guarantee is only fulfilled for the resistant samples, i.e., the majority group. For the sensitive samples, only the Summation score yields *valid* sets, i.e., sets that contain the true class with the desired certainty. In contrast, the coverage for the True Class and Mondrian score is only 73% and 85%, respectively. The higher coverage for the Mondrian score might be explained by the fact that this score is explicitly designed to fulfill the CP guarantee for each individual class (cf. Section 4.3.2.1).

For binary classification, a valid prediction set contains either only the true class of a sample or both classes. Consequently, we investigated the fraction of single-class predictions, i.e., the CP *efficiency*. The average efficiency across drugs is 80% for the True Class score but only 68% for the Mondrian Score and 2% for the Summation score.

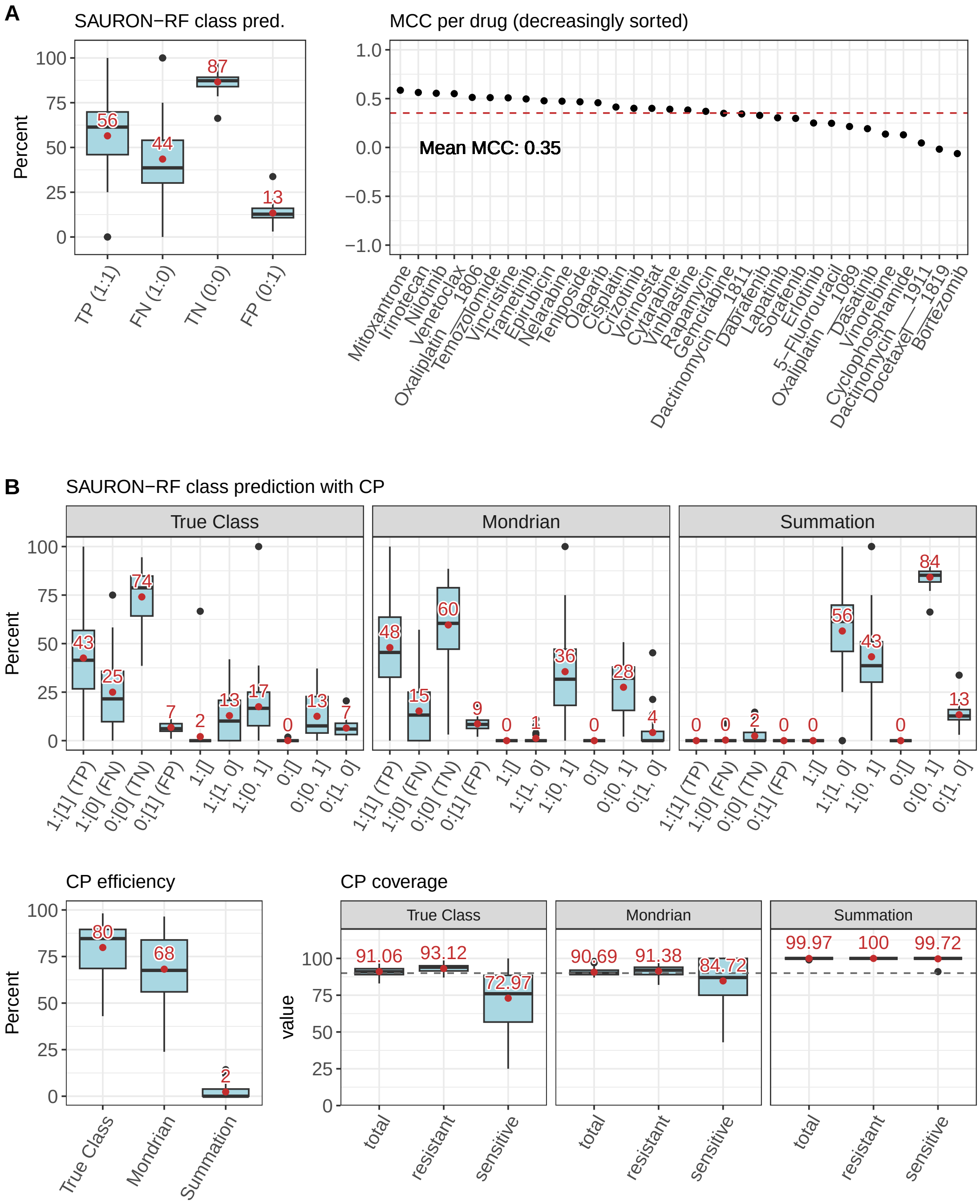


Figure 7.4: Classification results. This figure depicts the two-class classification performance of SAURON-RF without CP (A) and with CP (B) for 32 GDSC2 drugs using the respective test sets. Subfigure A depicts the performance of SAURON-RF without CP and shows the distribution of true positive (TP), false negative (FN), true negative (TN), and false positive (FP) predictions across drugs as well as Matthew's Correlation Coefficient (MCC) for each drug. Subfigure B shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency and coverage for each score are shown.

The very low efficiency of the Summation score indicates that almost all samples are predicted as a two-class set. Consequently, its coverage is very high since the true class is almost always contained in the predicted set. However, the predictions are not really informative since the score is rarely ever confident enough to predict a single class.
Lastly, we more closely investigated the single-class predictions for each score (cf. Figure 7.4B). We limit our discussions to the True Class and Mondrian score since the low efficiency of the Summation score does not allow drawing any conclusions on single-class predictions. Compared to SAURON-RF without CP, both the True Class and Mondrian score notably reduce the percentage of false positive (FP) and false negative (FN) predictions. The True Class score reduced FPs from 13% to 7% and FNs from 44% to 25%. The Mondrian score reduced FPs to 9% and FNs to 15%. Thus, some samples that were misclassified without CP are predicted as two-class sets by the CP scores, thereby avoiding these misclassifications.

However, along with the decrease in false predictions, a decrease in true predictions can also be observed: The True Class score reduced TPs from 56% to 43% and TNs from 87% to 74%. The Mondrian score reduced TPs to 48% and TNs to 60%.
Overall, the Summation score is outperformed by both other scores. The True Class score seems superior regarding the TN and FP predictions, while the Mondrian score seems superior regarding the TP and FN predictions.

Next, we investigated the regression performance of SAURON-RF with and without CP depicted in Figure 7.5. Without CP, the average test set MSE across drugs is 2.5, which is comparable to our findings in Chapter 6, and the Pearson correlation coefficient (PCC) is 0.56. With CP using the quantile regression score, we observe that the true IC50 is contained in 92% of predicted intervals on average. Thus, the CP certainty guarantee is again fulfilled. However, similar to the classification results, the coverage for sensitive samples of only 86% is lower than the coverage for resistant samples of 93%, which might again be caused by the underrepresentation of sensitive samples [267]. For regression, the CP efficiency can be defined as the width of the predicted interval, where more narrow intervals are more informative. Here, we consider the *relative* interval size, which is the size of the predicted interval divided by the size of the interval spanned by the IC50 range of the training samples. On average, the relative interval size across drugs is 0.5, indicating rather large intervals.

CMax viabilities: In our second analysis, we applied SAURON-RF to the CMax viability data. The corresponding plots can be found in Appendix Figures E.12 to E.15. Overall, the classification and regression performance without CP is comparable to the performance when using IC50 values: Averaged across drugs, a sensitivity of 64%, specificity of 76%, MCC of 0.35, and PCC of 0.51 were reached on the test set. The

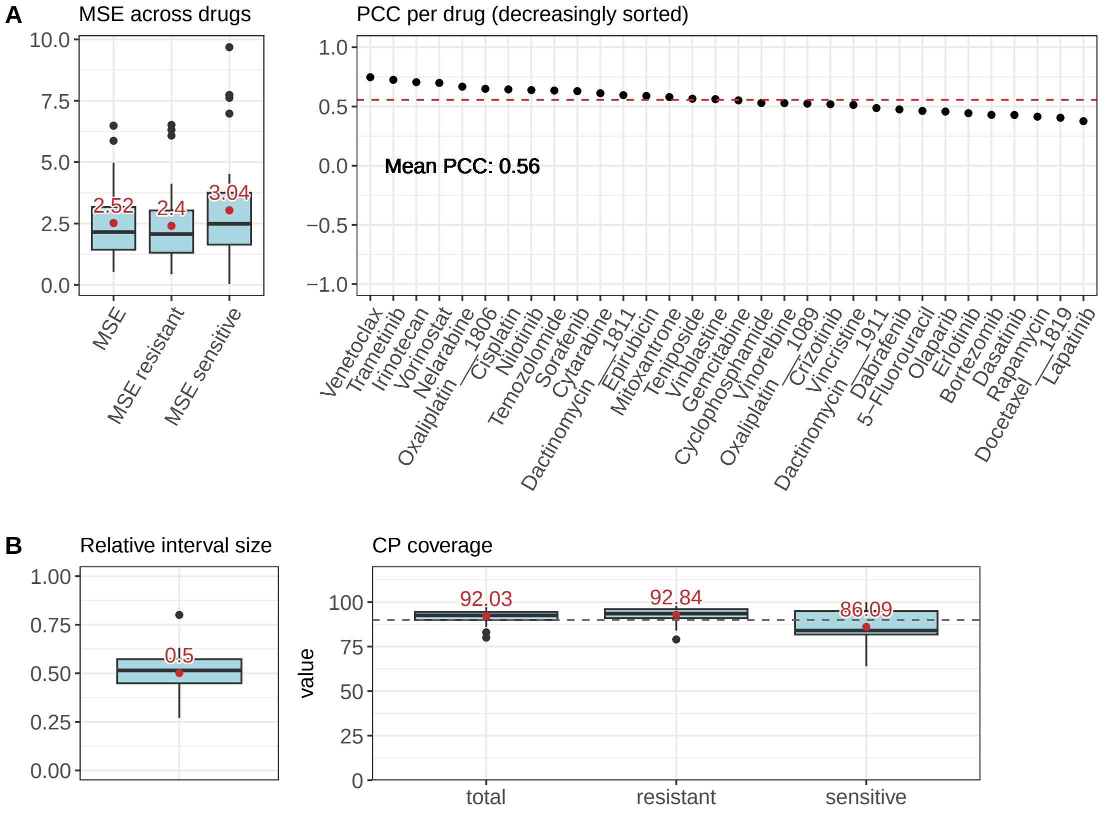


Figure 7.5: Regression results. This figure depicts the regression performance of SAURON-RF without CP (A) and with CP (B) for 32 GDSC2 drugs on the respective test sets. The top row depicts the mean squared error (MSE) and Pearson correlation coefficient (PCC) between the actual and predicted IC50 values for each drug obtained from SAURON-RF without CP. The MSE is also shown separately for the subsets of resistant and sensitive cell lines. Subfigure B shows the relative interval size across drugs after applying CP using the quantile regression score. Here, each predicted interval's size is divided by the interval size spanned by the response values in the training set of each drug. Additionally, the overall CP coverage and the coverage for the subsets of resistant and sensitive samples are shown.

MSE using CMax viabilities was 0.03, which is, however, not directly comparable to the MSE using IC50 values due to the different value ranges.

Class imbalances are also an issue when using CMax viabilities. However, in contrast to the IC50 data, where the minority class is always represented by the sensitive cell lines, there are 34% of drugs for which the resistant samples constitute the minority class when using the CMax viability as response measure (cf. Appendix Figure E.4). Nevertheless, the trends on CP coverage are again comparable to our observations for the IC50 data in the sense that the average coverage across samples is fulfilled by all scores and that the class-specific coverage tends not to be fulfilled for the minority class (cf. Appendix Figures E.14 and E.15). Also, the reduction of FP and FN predictions through CP was comparable to our IC50 analyses. Regarding regression efficiency, the relative interval size using CMax viabilities of 0.62 was even larger than the size of 0.5 for IC50 values.

7.2.1.2 Applying CP to the Deep Neural Network by Chiu et al.

To showcase the applicability of our CP framework to approaches other than SAURON-RF, we applied it to a modified version of a deep neural network (DNN) by Chiu et al. [292], which we also investigated in Chapter 5. Briefly described, their DNN uses gene expression and mutation data to predict the drug response of multiple drugs simultaneously. Since the DNN by Chiu et al. was originally developed for regression, we slightly adapted their approach to enable a classification and the use of CP: As activation function in the last layer of the network, we applied the sigmoid function such that the model outputs can be interpreted as class probabilities, specifically, the probability that a sample is *sensitive*. The probability of a sample being *resistant* is simply 1 minus the model prediction. As loss function, we applied the binary cross entropy (cf. Equation 4.64). Further details and visualizations of the results can be found in Appendix E.4.
Regarding the classification using discretized IC50 values, the model by Chiu et al. without CP fails to accurately predict the minority class of sensitive samples, as almost all samples are predicted as resistant (average sensitivity across drugs: 0). As extensively discussed in Chapter 6, this phenomenon is common for imbalanced drug sensitivity data across different types of ML models, which motivated us to develop SAURON-RF. With CP, the 90% certainty guarantee is achieved. Both the True Class and Mondrian score successfully remove some FN predictions. Additionally, the Mondrian score increases the fraction of TP predictions by around 20%. As discussed in Section 4.3.2.1, the Mondrian score has the unique property that the class with the highest prediction probability is not necessarily included in the predicted set. Consequently, single-class predictions of the Mondrian score may differ from the point predictions obtained without CP.
In a second analysis using the DNN by Chiu et al., we wanted to assess whether approaches other than SAURON-RF predict CMax viabilities with similar performance. Details and plots for these analyses are also provided in Appendix E.4. The average MSE across drugs is comparable for SAURON-RF (0.03, see Appendix Figure E.11) and the DNN (0.09). However, the mean PCC for the DNN is 0, indicating that this model cannot rank cell lines according to their response. In this regard, it is strongly outperformed by SAURON-RF with a mean PCC of 0.51 (see Appendix Figure E.11).

7.2.1.3 Three Classes

In the previous sections, we focused on the binary classification between sensitive and resistant cell samples. However, it is also common in drug sensitivity prediction to divide drug responses into three classes, e.g., *sensitive*, *intermediate*, and *resistant* [175, 176, 179, 384]. Such a more fine-grained division might reflect the drug response more accurately. It might also facilitate the differentiation between drugs that are highly sensitive from drugs that are effective but to a lesser extent, which would then be classified as *intermediate*. Lastly, a ternary class division might be helpful for machine learning to reduce the number of false sensitive and false resistant predictions.

Appendix Figures E.16 and E.17 show the results of SAURON-RF for a ternary classification using the CMax viabilities. As desired, misclassifications between the sensitive and resistant classes are rare, with only 9% of sensitive samples being falsely classified as resistant and only 6% of resistant samples being falsely classified as sensitive on average across drugs without CP. However, 37% of sensitive and 39% of resistant samples are falsely classified as intermediate. Nevertheless, 54% of sensitive and 55% of intermediate and resistant samples were correctly classified. The PCC (0.49) and MCC (0.3) are lower for the ternary compared to the binary setting (PCC: 0.56, MCC: 0.35). Additionally, the fraction of single-class predictions decreases strongly to 39% and 34% for the True Class and Mondrian scores, respectively, compared to 80% and 68% for the same scores in the binary setting. Consequently, we focus again on the binary classification setting for the following analyses.

7.2.2 Cell Line-Centric Analysis – Drug Prioritization

A major goal of personalized medicine is to use data from a tumor to recommend a list of suitable drugs ordered by their effectiveness. Here, we mimic this scenario known as *drug prioritization* for a given cell line as follows: First, we identify a subset of drugs where the response of the cell line is predicted to be sensitive via classification. We call these drugs *effective*. Second, we rank the effective drugs according to their regression predictions from most to least effective. As discussed above, comparing regression responses between drugs is not possible when using conventional measures such as IC50 and AUC. It is, however, possible when using our novel measure, the CMax viability, where smaller values indicate greater effectiveness.

Additionally, we can incorporate CP into the prioritization procedure: In the first step, we only consider those drugs for which CP returns a set containing solely the sensitive class. In the second step, we do not rank drugs according to their predicted CMax viability but according to the upper limit of the viability interval predicted by CP. The upper limit should serve as an indicator of the worst-case performance of the respective drug. An overview of the drug prioritization pipeline with CP is provided in Figure 7.3.

An example prioritization for a randomly chosen cell line (COSMIC-ID 1240154) is given in Figure 7.6 (see Appendix Figures E.22 to E.25 for further examples). First, we compare the actual discrete/continuous drug responses to the predictions of SAURON-RF: SAURON-RF without CP achieves an MCC of 0.66 and PCC of 0.9, indicating its effectiveness in identifying sensitive treatments and ranking treatments accurately.

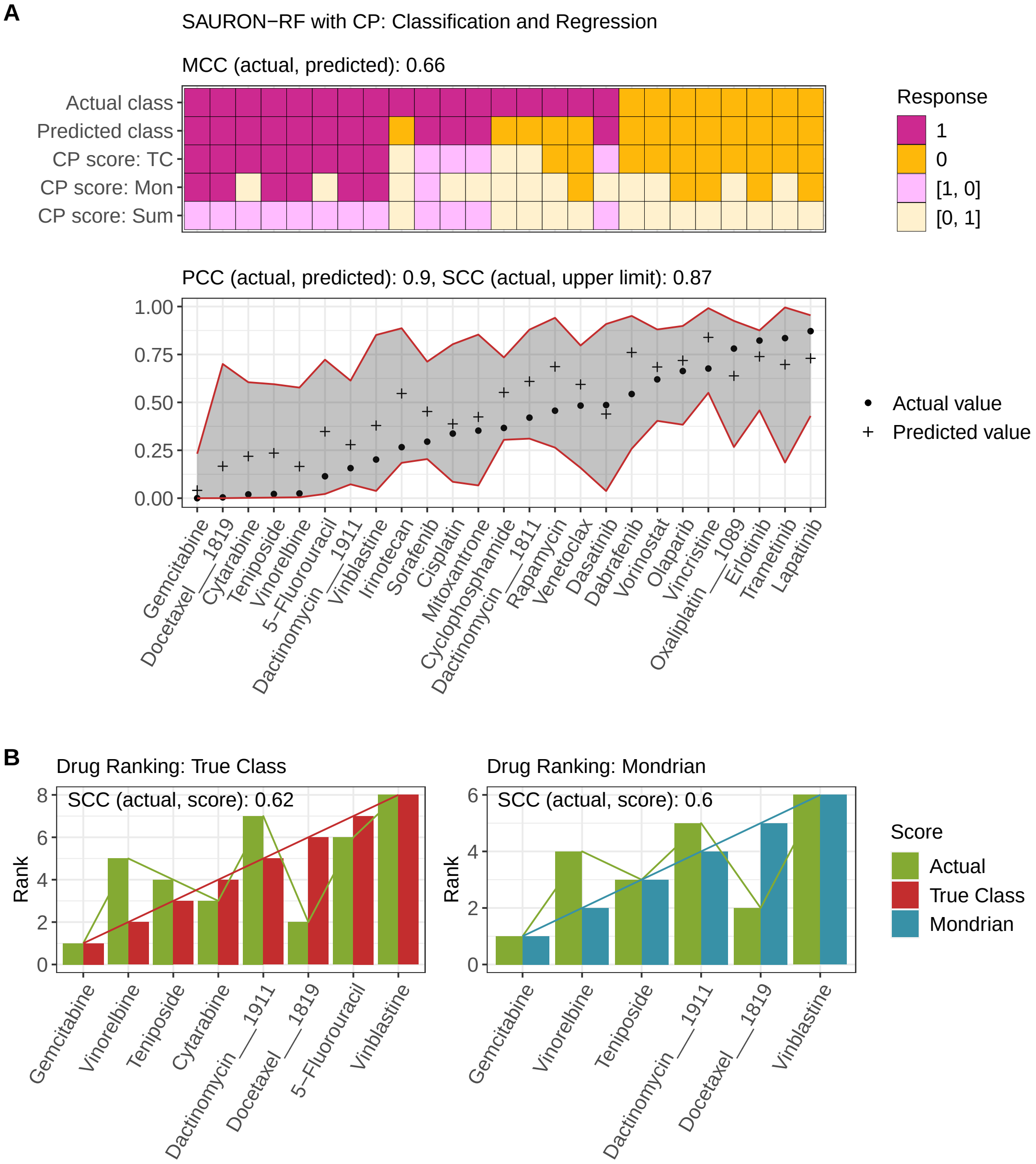


FIGURE 7.6: Example of drug prioritization. This figure depicts the results of drug prioritization for one exemplary cell line (COSMIC-ID: 1240154). Subfigure A shows the classification and regression results with and without CP (TC = True Class, Mon = Mondrian, Sum = Summation score). Subfigure B shows the prioritized drug lists generated from the CP results: For each score, drugs predicted to cause a sensitive response (depicted in the top plot of A) are sorted ascendingly by the upper limit of the predicted response interval (depicted in the bottom plot of A). Note that no prioritized list for the Summation score is shown since no drug was predicted to result in a sensitive response using this score. Further prioritization examples can be found in Appendix Figures E.22 to E.25.

However, five sensitive treatments are falsely predicted as resistant. Fortunately, using CP with either the True Class or Mondrian score removes all these FN predictions, albeit at the cost of losing some of the twelve TP predictions. The True Class score slightly outperforms the Mondrian score since it correctly identifies the eight most sensitive treatments as effective. In analogy to our previous analyses, the Summation score did not yield any single class predictions, so no ranking of effective drugs can be performed using this score.

The actual CMax viabilities and the upper limits of the predicted CP intervals are well correlated (Spearman correlation (SCC): 0.87), indicating an effective ranking. Still, the predicted intervals span a relatively wide range for most drugs as already observed in Figure 7.5.

The lower row of Figure 7.6 shows the prioritized drug lists for the True Class and Mondrian score, respectively. The rankings contain the same drugs in the same order, except for two drugs that are only present in the True Class ranking at positions four and seven.

Since this analysis is only an example for a single cell line, we investigated whether these findings extend to the entire set of test cell lines. As shown in Figure 7.7, SAURON-RF without CP again performs well for classification (MCC: 0.53, sensitivity: 71%, specificity: 81%) and regression (PCC: 0.81). Again, adding CP significantly reduces false classifications at the cost of losing some true predictions. Notably, the True Class score removes more false predictions and retains more true predictions than the Mondrian score. Regarding regression performance (cf. Figure 7.7B), the SCC between predicted and actual values without CP is 0.82, which is slightly larger than the SCC of 0.75 between the actual value and the upper limit of the predicted CP interval.

One crucial challenge of drug prioritization is to not include resistant treatments in the suggested lists. Thus, we evaluated the precision of the lists, i.e., the percentage of contained TP predictions (cf. Figure 7.7C). The median precision for SAURON-RF without CP is 76%, which is outperformed by both the True Class (92%) and Mondrian (83%) scores. The lists generated by the True Class and Mondrian score contain the actual most effective drug in 75% and 56% of cases, compared to 90% for SAURON-RF without CP. However, the average viability difference between the actual most effective drug and the drugs predicted to be most effective by both the True Class and Mondrian score is below 0.1 for most of cell lines (62% for True Class, 56% for Mondrian, cf. Figure 7.7D). This means that both scores tend to first recommend drugs that are similar in effectiveness to the actual most effective drug.

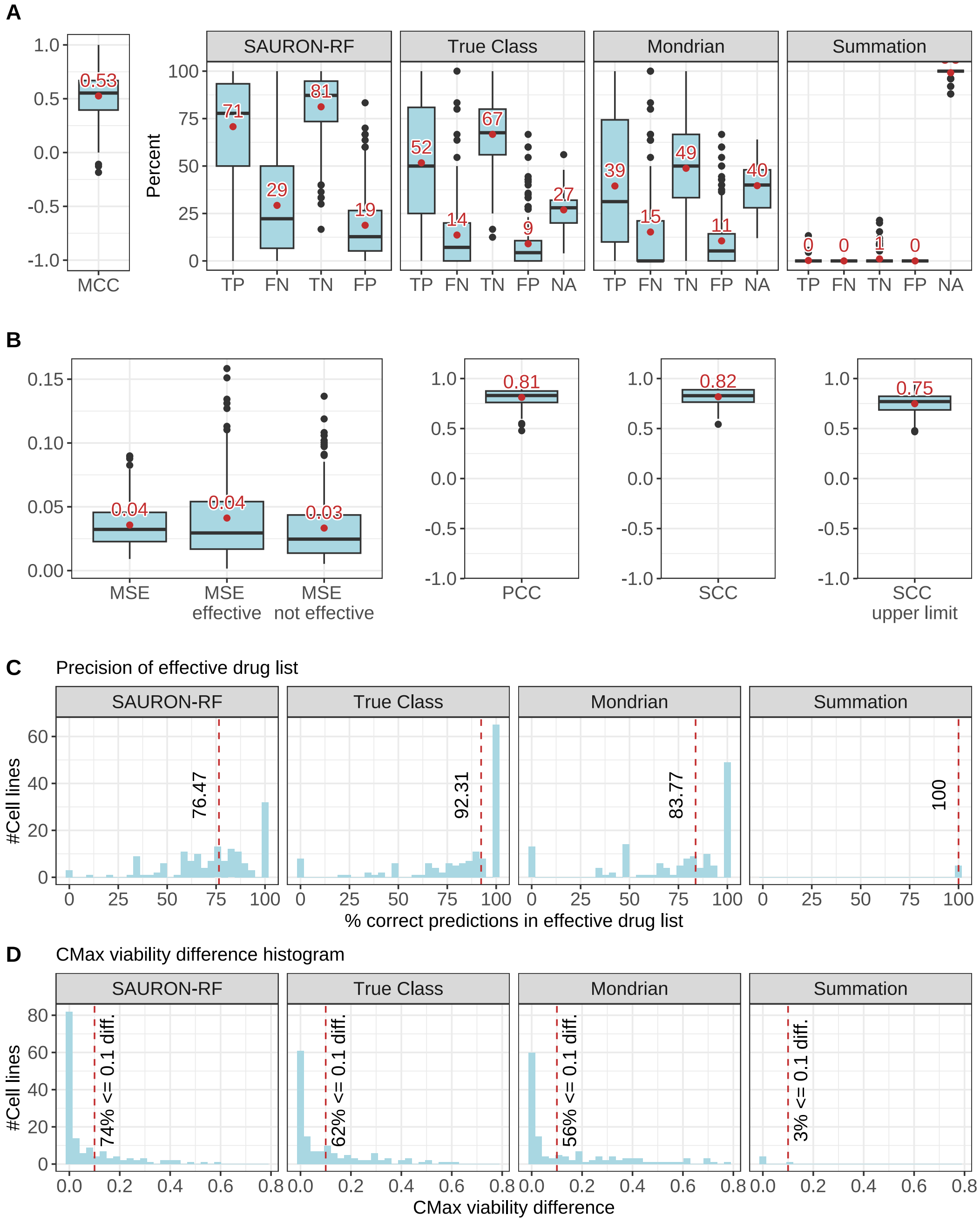


FIGURE 7.7: Prioritization results. Subfigure A shows the classification performance of SAURON-RF. For the CP scores, single-class predictions are shown as TP, FN, TN, and FP, while all types of two-class predictions are summarized and denoted as NA. Subfigure B shows the regression performance. Here, the average MSE is shown for all drugs and for the subsets of effective and ineffective drugs. Additionally, the PCC and the SCC between the actual and predicted values or the upper limit of the CP interval are shown. Subfigure C depicts the precision, i.e., the percentage of TPs in the prioritized lists generated using SAURON-RF without and with CP. The dashed red line marks the median precision. Subfigure D shows the CMax viability difference between the actual most effective drug and the drug predicted to be most effective. The dashed red line marks a difference of 0.1.

7.3 Discussion

In this chapter, we addressed two major challenges of making personalized treatment recommendations with ML: (1) providing reliability estimates and certainty guarantees for drug sensitivity prediction and (2) performing drug prioritization for a given cell line. To address the first challenge, we developed a CP pipeline in Python. Our pipeline can be applied to any classification and regression approach that provides a notion of prediction (un-)certainty to generate reliable predictions at a user-specified certainty level α. Our evaluations using SAURON-RF show that predictions generated with CP achieve the desired certainty guarantees. Such guarantees are highly desirable to enhance the trustworthiness of ML, e.g., in a clinical setting. Additionally, CP reduces both false positive and false negative predictions. Minimizing false predictions is another crucial factor for the practical application of ML, as mistakes – such as suggesting ineffective drugs to treat a patient – may have severe consequences.

Additionally, compared to conventional point predictions, CP allows a more nuanced assessment of drug response: By predicting sets or intervals, CP provides a range of potential treatment responses. In a clinical setting, this could help physicians to assess the uncertainty associated with the predicted treatment response for individual patients. This may improve risk management, as potential variations in a patient's response can be better anticipated, and treatment plans can be adjusted accordingly. Furthermore, cases where the predicted set or interval is large indicate that the model cannot provide a more precise estimation of drug response for the specific sample. Even though we would like such cases to be rare, we would certainly prefer a model that can express its uncertainty rather than a model that always makes precise but frequently incorrect predictions.

Besides reliability estimation, the second challenge addressed in this chapter is drug prioritization. To allow the prioritization of drugs for a given cell line, we developed a novel drug sensitivity measure called CMax viability that is comparable across drugs. Our measure is based on clinically feasible treatment concentrations and may, thus, facilitate translating ML-based findings into clinical application.

In our analyses, we combined drug prioritization with CP to generate a reliable ranking of effective drugs: In the classification step, we ensure to only rank drugs that can confidently be predicted as effective for the cell line of interest. While SAURON-RF without CP results in 19% FP predictions, i.e., drugs that are falsely predicted as effective, the addition of CP removes 52% of these FPs. Consequently, 92% of drugs in the prioritized lists are correctly predicted to be effective. Furthermore, the actual most effective drug was contained in our lists in 75% of cases. In the regression step, we rank the identified drugs based on response intervals derived from quantile regression. Our predicted rankings correlate well with the actual rankings. Furthermore, the drug predicted as most effective by our ranking is similar in effectiveness to the actual most effective drug. Based on these findings, we are convinced that our CP and prioritization pipeline could be a valuable tool for medical decision support.

Nevertheless, several aspects of our pipeline could potentially be improved to enhance performance and emulate the real-life application scenario more accurately: First, we only focus on cell line data here. As discussed in Chapter 3.1, model systems such as patient-derived xenografts or organoids may represent tumor characteristics more accurately [151]. Additionally, we only used gene expression data to characterize cancer cell lines. Even though gene expression is assumed to be the most informative data type for drug sensitivity prediction [23, 176], additional data types such as mutation or copy number variation data, biological pathways, or protein interaction networks may enhance model performance and interpretability [49, 177, 294]. Moreover, we could incorporate a priori knowledge in our prediction models, e.g., known response biomarkers [37] or drug targets [293].

In our analyses, we investigated three CP scores for classification but only one for regression. Since the predicted regression intervals were relatively large, considering other scores may alleviate this issue. Additionally, the applicability of CP for classification in our pipeline is more straightforward than for regression: The class probabilities that we use to perform CP for classification can easily be obtained from most ML algorithms (cf. Section 4.2) and this functionality is provided in most Python or R packages for ML. However, this is not the case for quantile regression, which we used to perform CP for regression. Thus, investigating further CP scores or other CP-related techniques, as discussed by Vazquez and Facelli [385], may make reliability estimation more easily applicable and might further improve results.

While we extensively investigated model performance, drugs with especially high class imbalance were removed from our analyses. Even though we showed that SAURON-RF can effectively counteract imbalances and that CP can reduce misclassifications, such datasets remain challenging for ML. However, for a real-life application of ML-based treatment recommendation systems, even treatments that are only effective for a very small number of cases should be accurately prioritized. Consequently, such drugs require further investigation.

Lastly, similar to the previous chapters, we focused on training drug-specific prediction models. Consequently, our framework cannot be used to prioritize previously unseen drugs, e.g., newly developed ones. In the next chapter, we introduce models that can make predictions for multiple drugs, even previously unseen ones, by incorporating a drug representation in the model input. Additionally, we predict drug responses not only for monotherapies but also consider two-drug combination therapies. Combination therapies are often preferred over monotherapies to treat cancer as they can increase treatment effectiveness and decrease the risk of developing resistances [41]. The next chapter will also introduce a modified version of the CMax viability for drug combinations to prioritize both single- and multi-drug treatments.

Chapter 8

Predicting Drug Responses for Combination Therapies

In the previous chapters of this thesis, we were concerned with predicting the drug response of cell lines treated with single drugs, i.e., monotherapies. However, as discussed in Chapter 2.2.4, combination therapies are frequently preferred over monotherapies for cancer treatment due to increased efficacy and a decreased risk of treatment resistance [41]. For monotherapy, large cell line panels such as the *Genomics of Drug Sensitivity in Cancer* (GDSC) database [49, 129] have been available for more than a decade. More recently, large data resources have also become available for drug combination screens. In 2019, the DrugComb data portal was introduced [46, 50], accumulating harmonized drug screening results from different sources. Currently, a total of 37 datasets are available in DrugComb [46].

To date, drug combination datasets such as DrugComb are still underused for drug sensitivity prediction. Instead, most models trained on this data focus on synergy prediction. Here, the goal is to predict whether the effect of a certain drug combination is greater (or smaller) than expected based on the drugs' individual effects. To this end, the synergistic (or antagonistic) potential of two compounds for a cell line is typically quantified using synergy scores [194, 386]. Prominent examples are the Loewe [185], Bliss [186], HSA [187], and ZIP [184] synergy scores, which we discussed in Section 3.3.1.

Undoubtedly, estimating the synergistic potential of compound combinations through synergy scores can be valuable for identifying promising combination treatments to undergo more detailed screening. Additionally, it may the support the development of novel compounds explicitly designed to work in synergy with others. However, even though synergy score prediction is sometimes motivated as a step toward achieving personalized treatment recommendations [194, 287], we are convinced that synergy scores have shortcomings that debilitate their usefulness for this application, which will be outlined in Section 8.1.

Thus, instead of relying on synergy scores, we advocate exploring other strategies to estimate the effectiveness of combination treatments. For predicting drug combination sensitivity, several machine learning (ML) models that do not rely on synergy scores have been published: Malyutina et al. [183] and Zagidullin et al. [50] trained cell-line specific models that predict CSS (Combination Sensitivity Score) values, a sensitivity measure for two-drug combination therapies, we described in Section 3.3 [183]. However, the CSS score is an aggregated measure based on drug-specific AUC values that combine information across the entire tested concentration range to quantify sensitivity. Thus, like the AUC for monotherapies [40], it depends strongly on the chosen concentration ranges and is not comparable across compounds (cf. Section 3.2.4).

Instead of predicting measures like the CSS that are often aggregated over multiple concentrations, an alternative is to make dose-specific predictions of drug response. Here, the response is typically measured as the effect of the treatment on cell growth/viability, e.g., through relative inhibition values (cf. Section 3.3.1). For monotherapies, this approach has already been explored by Rahman and Pal et al. [387, 388]. For combination therapies, Zheng et al. [46] trained a CatBoost model that predicts the relative inhibition of two drugs at given concentrations for a given cell line. Similarly, comboFM by Julkunen et al. [389] employs higher-order factorization machines to predict relative cell growth at specific doses.

A drawback of all combination prediction approaches mentioned above is that they are not applicable to make predictions for previously unseen cell lines, i.e., cell lines that were not included in the training data: Malyutina et al. [183] and Zagidullin et al. [50] trained cell line-specific models, where each model can only make predictions for the one cell line it was trained on. In contrast, Zheng et al. [46] and Julkunen et al. [389] employ a one-hot encoding of cell lines and drugs in the model input such that both have to be known during training already. Thus, these models are difficult to apply for personalized treatment recommendations where predictions should be made for a previously unseen patient (cell line). According to Codicè et al., this setting is frequently overlooked or insufficiently evaluated in ML-based drug response prediction [313].

In this chapter, we present ML models for predicting drug combination sensitivity that do not rely on synergy scores and can make predictions for previously unseen cell lines, thereby mimicking the personalized treatment scenario. Instead of predicting an aggregated measure of treatment response, our models predict the relative inhibition at certain treatment concentrations specified in the model input. To the best of our knowledge, we are the first to develop models that can make dose-specific predictions of drug combination sensitivity for previously unseen cell lines (cf. Table 8.1 in Section 8.3.4 later in this chapter). Our model design allows the reconstruction of various measures of drug sensitivity or synergy from the model predictions, including dose-response curves and matrices, as well as IC50 values and synergy scores.

Using data from DrugComb, we investigate the prediction performance of different ML algorithms and analyze the benefit of including different drug characterizations as well as information on drug targets in the model input. Additionally, we assess the reconstruction of mono- and combination sensitivity measures from the model predictions. Finally, we show how our models can be applied to perform drug prioritization for mono- and combination therapies. To this end, we extended our novel drug sensitivity measure, the *CMax viability* (cf. Chapter 7), to be applicable to combination treatments.

Author Contributions

The content and text of this chapter are based on the following manuscript and corresponding preprint, which were both written by me:

Eckhart, L., Lenhof, K., Herrmann, L., Rolli, L. M. & Lenhof H. P. (2025). *How to Predict Effective Drug Combinations - Moving beyond Synergy Scores.* Accepted for publication in iScience. DOI (iScience pre-proof): 10.1016/j.isci.2025.112622, DOI (bioRxiv preprint): 10.1101/2024.11.22.624812

The study was conceived by me, Kerstin Lenhof, and Hans-Peter Lenhof. I designed the study, was involved in the data processing, implemented the machine learning models, conducted the experiments, and drafted the manuscript. These steps were supervised by Kerstin Lenhof and Hans-Peter Lenhof. All authors analyzed the data, evaluated the results, and reviewed the manuscript.

8.1 Challenges of Synergy Scores

In this section, we discuss the challenges associated with synergy scores, especially for deriving personalized treatment recommendations. These challenges motivated us to develop ML models that predict the drug response of combination therapies without relying on synergy scores, which will be discussed afterward. While a detailed explanation of synergy scores and their calculation is provided in Section 3.3.1, we briefly reiterate the most important concepts below.

The idea behind synergy scores is to measure the synergistic or antagonistic potential of two (or sometimes more) compounds for a given cell line by comparing their experimentally measured combined effect on cell survival to the expected effect obtained from a baseline model that assumes no synergism or antagonism [184]. The baseline model is derived from monotherapy data of both compounds. It estimates the combined effect of the two compounds at the same concentrations as tested in the actual combination screening. For each concentration combination, the baseline and actually measured treatment responses are then subtracted from each other. Lastly, the results are averaged over all concentration combinations to obtain a final synergy score [43]. Prominent examples of synergy scores that differ solely in their computation of the baseline are the Loewe [185], Bliss [186], HSA [187], and ZIP [184] scores. For each score, values > 0 indicate synergism, and values < 0 indicate antagonism. A detailed description of the scores can be found in Section 3.3.1.

Estimating the synergistic potential of compound combinations through synergy scores is certainly valuable for identifying promising combination treatments to undergo more detailed screening or developing novel compounds explicitly designed to work in synergy with others. However, there are known limitations of synergy scores, which have been summarized and extensively discussed in a review by Vlot et al. [43]. They also investigated the agreement and across-batch reproducibility of four synergy scores (Loewe, HSA, ZIP, Bliss) using a large-scale drug combination dataset. Their findings can be summarized as follows: Firstly, each synergy score is based on a set of model assumptions that differ between scores and may also be violated by real-world data [44, 45]. A detailed explanation of score-specific assumptions is provided in Section 3.3.1. For example, both the Loewe and ZIP score require fitting dose-response curves of a certain shape to the monotherapy data. The Loewe score furthermore requires both drugs to have the same minimum and maximum effect as well as a constant potency ratio (cf. Equation 3.19) [43]. In comparison, the Bliss score assumes that the combined effect of two non-interacting drugs can be modeled through statistical independence.

These varying model assumptions might explain the moderate to low correlation observed by Vlot et al. between the different scores calculated on the same data. Furthermore, while complete disagreement (synergism vs. antagonism) between scores was rare, Vlot et al. identified several scenarios where scores are likely to disagree, which could typically be retraced to a violation of model assumptions. Interestingly, although Vlot et al. report a strong correlation between the measured drug responses in terms of viability, the derived synergy scores are comparatively difficult to reproduce in replicated experiments.

Based on these findings, Vlot et al. advocate against the automated analysis of large-scale data using individual synergy scores. Instead, they recommend carefully investigating individual dose-response curves to decide which of the existing scores is applicable, considering the score-specific assumptions.

We agree with these conclusions of Vlot et al. Additionally, we identified further aspects that make synergy scores difficult to use and interpret, especially for deriving personalized treatment recommendations: A methodological criticism of synergy scores is that

they are an aggregated measure of treatment response over the tested concentration ranges. Choosing meaningful concentration ranges is especially challenging for experimental drugs but crucial to drawing meaningful conclusions for personalized medicine. In Chapter 3.2.4.1, we have discussed that the screened concentration ranges in the GDSC database do often not correspond well to clinically feasible treatment concentrations (cf. Appendix Figure B.1) and similar observations can also be made for the DrugComb database (cf. Appendix Figure B.2). Consequently, it is questionable how expressive synergy scores calculated on (in part) unrealistically small/large concentrations are. Additionally, it is assumed that high synergy between compounds may only manifest in relatively narrow dose windows [390]. Consequently, aggregating information over the entire concentration ranges may fail to capture such windows.
Another major factor that hampers the use of synergy scores for treatment recommendation is that a high synergy between two compounds solely implies that the combination treatment is more effective than expected from the monotherapy responses of these two compounds. However, it does not guarantee an overall high effectiveness (in terms of large relative inhibition) of the combination treatment [46]. Likewise, in a clinical setting, combination synergy is not the most conclusive factor for treatment success: Palmer and Sorger found that the benefit of most combination therapies in clinical trials can be explained by independent drug action rather than synergy [391].

Based on these drawbacks of synergy scores in general and for treatment recommendation in particular, our models described in the following focus on sensitivity prediction instead. While there are numerous methods for predicting synergy [194, 386], sensitivity prediction of drug combinations is relatively underexplored, especially when the goal is to make predictions for previously unseen cell lines as we have outlined at the beginning of this chapter.

8.2 Materials and Methods

In the following, we describe our approach for predicting drug combination sensitivity. Instead of predicting an aggregated measure of treatment response, our models predict the relative inhibition at certain treatment concentrations specified in the model input. We first describe the dataset used for model training. Next, we describe our model design, including the three investigated ML algorithms and four different ways to represent drugs and cell lines in the model input. Afterward, we describe how dose-response curves and matrices can be reconstructed from the model predictions. Finally, we explain how sensitivity measures can be derived from these curves/matrices. In this context, we also introduce a novel sensitivity measure for combination therapies, the *combination CMax viability*.

8.2.1 Dataset

For the analyses presented in this chapter, we considered three types of data: (1) drug response data of cell lines screened with individual drugs or two-drug combinations, (2) drug characterizations based on their molecular structures, and (3) cell line characterizations based on gene expression.

The normalized gene expression values for 17,419 genes were obtained from the GDSC database Release 8.3, which we described in detail in Section 3.4. The other two data types and the data processing are discussed in the following.

8.2.1.1 Drug Response Data

Drug screening data for our analyses was obtained from the DrugComb database v.1.5 described in Section 3.5. We employed the DrugComb API (https://api.drugcomb.org/) to download a list of all cell lines contained in DrugComb with their corresponding COSMIC IDs, a list of all drugs, and all available dose-response data.

To assign the correct cell line and drug(s) to each dose-response experiment, we downloaded the core database from https://drugcomb.org/download, which provides a unique identifier for each experiment. Based on this information, each database entry from DrugComb can be represented as follows:

$$(cell\_line, drug\_row, drug\_col, conc\_row, conc\_col, inhibition)$$

Here, *cell_line* is the COSMIC ID of the investigated cell line, and *drug_row* and *drug_col* are the names of the tested drugs. The entries *conc_row* and *conc_col* are the micromolar concentrations of the tested compounds. For monotherapies, one of the drug names is set to $NULL$, and the corresponding concentration is set to 0. Finally, *inhibition* denotes the relative inhibition measured after administration of the denoted drug concentration(s).

In the previous chapters, we used the relative viability instead of the relative inhibition as measure of drug response. As discussed in Section 3.3.1, the relative inhibition can simply be calculated as 1 minus the relative viability. In the DrugComb database, the resulting values are additionally multiplied by 100.

Typically, the resulting relative inhibitions are in the range $(-\infty, 100]$, where values < 0 indicate that the treatment increases cell growth and values > 0 indicate a reduction in growth. Values > 100 should generally not occur since they would, paradoxically, indicate that the treatment killed more cells than were present initially. However, in cases where a treatment killed all cells, values > 100 can rarely occur (here: 0.18% of the data). Section 3.2.2 discusses this phenomenon for relative viabilities. Briefly summarized, such values are caused by differences in the measured luminescence signals of an empty well (called *positive control* in Section 3.2.2) and a well which initially contained live cells that were then killed by the treatment in the in vitro drug screening.

Next, we removed the following entries from the dataset:

- poor quality entries as defined by the authors of DrugComb [46], i.e., entries with $inhibition < -200$ or $inhibition > 200$
- entries where the concentration of all tested drugs is 0 ($conc\_row = conc\_col = 0$)
- entries where the corresponding cell line had no COSMIC ID or no gene expression data provided in the GDSC database

Additionally, we converted entries where $drug\_row$ and $drug\_col$ denote the same drug into monotherapies by summing the respective treatment concentrations and setting $drug\_col$ to $NULL$:

$$(cell\_line, drug\_row, NULL, conc\_row + conc\_col, 0, inhibition)$$

Cases where two different drugs are provided but only one has a concentration > 0 were modified to denote a monotherapy by replacing the drug with concentration 0 with $NULL$.

Afterwards, the inhibition values of all replicates involving the same cell line, the same drug(s), and the same concentration(s) were averaged. Given the inhibition range of $[-200, 200]$, the average standard deviation between replicates was small (mean: 7, median: 5). Lastly, we log1p-normalized ($log1p(x) = log(x + 1)$) the concentration values in $conc\_row$ and $conc\_col$.

To keep the dataset size manageable, we limited the number of investigated drugs: For each drug d, we counted the number of entries E_d where $drug\_row = d$ or $drug\_col = d$. Next, we only kept entries consisting exclusively of those 265 drugs for which $E_d \geq 10,000$ (cf. Appendix Table F.1). The final dataset consists of 5,291,424 entries covering 947 cell lines, 265 drugs, and 9,535 drug combinations.

Additionally, the CMax concentrations for 77 of the investigated drugs were obtained from Liston and Davis [174]. The CMax denotes the peak plasma concentration after administering the highest clinically recommended dose of a drug [174]. In Chapter 7, we employed CMax to derive a novel drug sensitivity measure for monotherapies called the *CMax viability* (cf. Section 7.1.2). The CMax viability measures the relative viability after treating a cell line with the CMax concentration of a drug. It enables comparing the effect of different drugs on the same cell line. In Section 8.2.5, we will derive an extension of the CMax viability for combination therapies called the *combination CMax viability*. Based on the (combination) CMax viability, we can compare the effectiveness of both mono- and combination therapies for the same cell line, which we will analyze later in this chapter.

8.2.1.2 Drug Properties

For the representation of drugs in the input of our models, we investigated different settings, which will be discussed below. In particular, we investigated two types of drug features:

- **MACCS fingerprints** [392]: A MACCS (Molecular ACCess System) fingerprint is a binary vector where each entry corresponds to a molecular substructure, e.g., a functional group that may be present in a drug molecule. The respective bit is set to 1 if the corresponding substructure is present in the drug molecule at least once and 0 otherwise.
- **Physico-chemical drug properties** [393]: These properties describe different characteristics of a drug, such as the molecular weight, number of valence electrons, or the logP value that measures lipophilicity.

Both MACCS fingerprints and physico-chemical properties were calculated using RDKit [394] based on the SMILES drug representations provided by DrugComb. The SMILES (Simplified Molecular Input Line Entry System) format represents a (drug) molecule as a human- and machine-readable character string that encodes the molecule's atoms and bonds [395]. We removed all properties that showed no variation across the investigated 265 drugs, resulting in MACCS fingerprints of length 162 and 182 physico-chemical properties.

Additionally, 735 drug target molecules for the investigated drugs were obtained from DrugComb. For our analyses, we only considered those 290 molecules targeted by at least five drugs in our dataset.

8.2.2 Model In- and Outputs

We train multi-drug models that predict the relative inhibition for a given cell line being treated with given concentrations of one or more drug(s). The model inputs comprise cell line features based on gene expression, a representation of the applied drugs, and the corresponding drug concentrations.

To characterize cell lines in the model input, we performed a principal component analysis (PCA, cf. Section 5.1.4) on the gene expression values of the training cell lines and used the first 300 principal components (PCs) as cell line features. This dimension reduction method and feature number performed well in our benchmarking in Chapter 5. The feature coefficients computed on the training data were used to project the test cell lines into the same 300-dimensional space.

In addition to the cell line features, we investigated four different settings for the encoding of drugs in the model input, which are visualized in Figure 8.1:

Setting 1 (OneHot):

In this setting, no drug properties are included. Instead, a 265-dimensional encoding of drugs is used. Each feature corresponds to one of the 265 drugs in our dataset. If a drug is part of the current entry (i.e., the currently considered combination of a cell line, treatment drug(s), and the respective concentration(s)), its feature is set to the corresponding log1p-normalized treatment concentration. Otherwise, it is set to 0.

Setting 2 (OneHotTar):

This setting uses the same concentration encoding as Setting 1 but additionally includes 290 features representing drug target molecules. Each of these features is set to the number of drugs in the current entry that target the corresponding molecule. Since DrugComb provides only data on monotherapies and two-drug combinations, the maximum value a target feature can have is 2 if it is targeted by both drugs in a two-drug combination entry. Note also that one drug can target more than one molecule.

Setting 3 (MACCS):

In this setting, each drug is represented by a 162-dimensional binary MACCS fingerprint. Additionally, one input feature for each drug is needed to denote its treatment concentration. Consequently, this setting uses $2 \cdot 162 + 2 \cdot 1 = 326$ drug features. For monotherapies, one of the fingerprints and the corresponding concentration are set to 0.

Setting 4 (PhysChem):

This setting is similar to Setting 3 but replaces the MACCS fingerprint with 182 numerical physico-chemical descriptors that denote different properties of the respective drugs. Consequently, this setting uses $2 \cdot 182 + 2 \cdot 1 = 366$ drug features. For monotherapies, one set of properties and the corresponding concentration are set to 0.

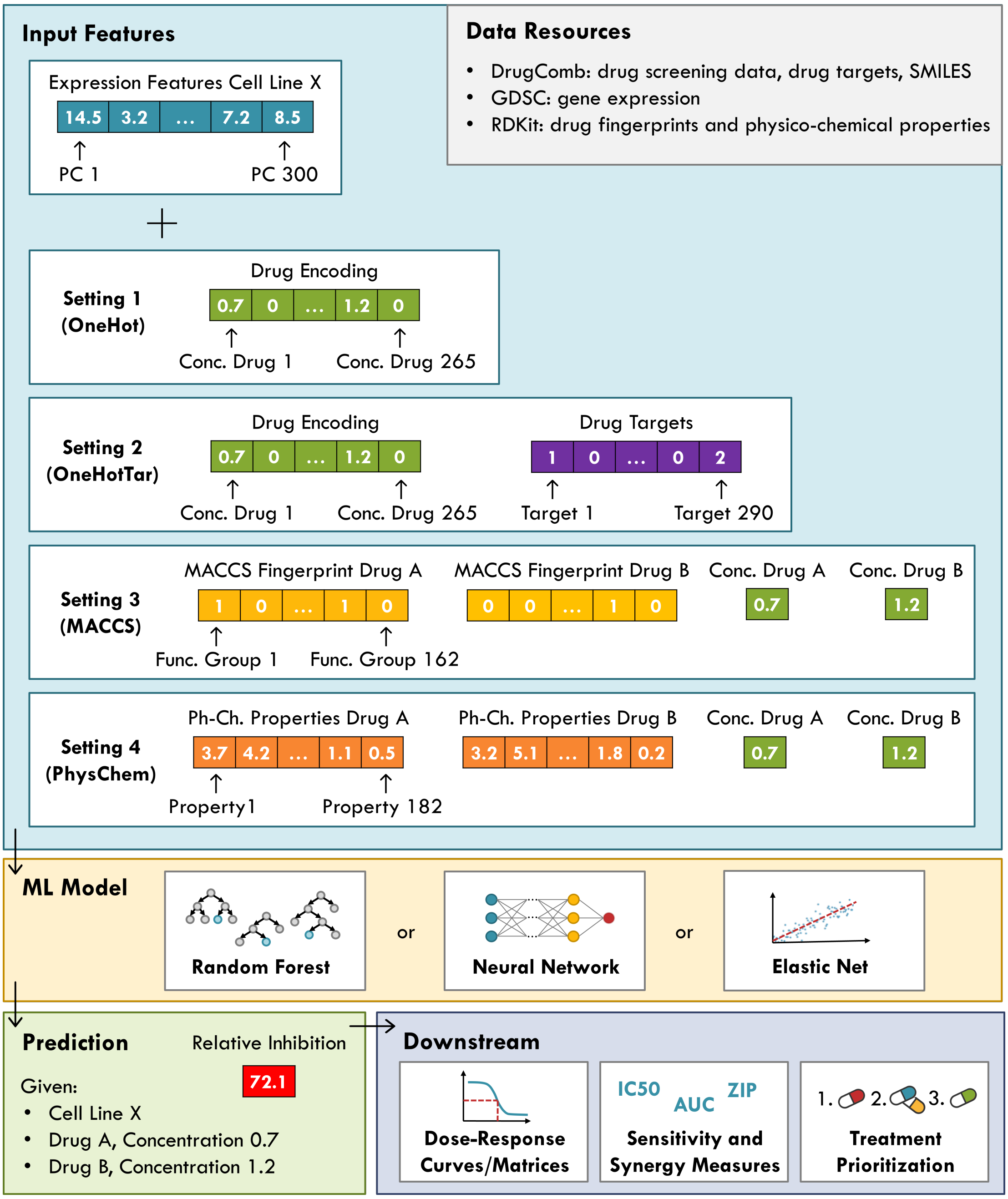


FIGURE 8.1: Prediction pipeline. This figure summarizes our pipeline for predicting relative inhibitions. The large blue box depicts the different types of input features and representations we investigated. The gray box at the top right lists our data resources. The yellow box shows the different ML algorithms we used. The green box at the bottom depicts the model output, i.e., the relative inhibition for a given cell-drug-drug combination at the defined treatment concentrations. Lastly, the purple-gray box shows downstream analyses that can be performed based on the model predictions.

Depending on the desired application, the different settings provide different benefits: Settings 3 and 4 allow making predictions for arbitrary drug molecules, given that their MACCS fingerprint or physico-chemical properties are known. Consequently, the resulting models can be used to make predictions for previously unseen, e.g., newly developed compounds. In contrast, models derived from Setting 1 and 2 are limited to those 265 drugs that are present in our dataset and hence encoded in the input. However, these models can not only make predictions for single drugs and two-drug combinations but even for treatments using three or more drugs simultaneously. While three-drug combination therapies have already been approved for cancer treatment by the United States Food and Drug Administration (FDA) [108], DrugComb does not provide such data.

8.2.3 Machine Learning Algorithms

We investigate the predictive performance of three ML algorithms: neural networks, random forests, and elastic nets. Both neural networks and tree-based methods are commonly used for synergy prediction [386]. In our benchmarking in Chapter 5, we found, however, that tree-based methods and elastic nets frequently outperform neural networks in predicting drug responses.

8.2.4 Model Training and Testing

After filtering and processing the data as described above, we randomly divided the remaining cell lines into a training set (80% of cell lines) and a test set (20%). Since multiple data entries exist for each cell line (screening of different drugs/drug combinations and different concentrations), the final training data consists of all entries involving a cell line from the training set (3,741,209 entries). The final test data contains all remaining entries (1,550,215), i.e., all entries involving a cell line from the test set. This splitting ensures that the test performance is always evaluated on cell lines that were unseen during model training, thereby mimicking the scenario of making predictions for a previously unseen patient. In contrast, the same drugs and drug combinations can occur in both the training and test data.

On the training data, we performed a 5-fold cross-validation (CV) to determine the best-performing hyperparameters of each ML model (see Appendix Table F.2). The CV folds were generated by randomly dividing the training cell lines into five disjoint folds and assigning all entries involving a certain cell line to the corresponding fold. Since the number of available entries per cell line differs, the size of CV folds varies slightly between 644,308 and 857,361 entries. For the hyperparameter combination with the smallest mean absolute error (MAE, cf. Section 4.3.1) averaged across all five folds, one final model is trained on the complete training data, and its performance is evaluated on the test data.

For the models using one-hot encodings (Setting 1 and Setting 2), each drug has a designated input node. This is not the case for the models using drug features (Setting 3 and Setting 4), where swapping the features and concentration of the first drug with those of the second drug represents the same treatment but results in changes in the input representation (cf. Figure 8.1). However, the model output should not depend on the order of the drugs in the input, i.e., it should not depend on whether drug features of a drug A in the input vector are located in front of or behind those of a drug B. Therefore, each original sample is included twice in the datasets for Settings 3 and 4. These duplicate samples differ only in the order of the drug features and concentrations: once in the order A-B, once in the order B-A. In Section 8.3, we investigate the impact on model performance when models are trained using the duplicated versus non-duplicated data. The test performance is always evaluated on the duplicated entries.

8.2.5 Fitting Dose-Response Curves and Computing Sensitivity Measures

This section describes how the relative inhibitions predicted by our models can be further processed to reconstruct dose-response curves for monotherapies and dose-response matrices for combination therapies (cf. Figure 8.2 and Chapter 3). Based on these curves/matrices, various measures of drug response can be derived. To this end, we first converted the (actual and predicted) relative inhibitions into relative viabilities by subtracting the relative inhibitions from 100 and dividing the result by 100. Additionally, we clamped viabilities to $[0, 1]$. Note that we report relative viabilities in the range $[0, 1]$ rather than range $[0, 100]$ to keep the results consistent and comparable to our analyses in Chapter 7.

To perform the curve-fitting for monotherapies, we employed a three-parametric logistic function from the *drc* R-package (cf. Equation 3.4 with $\alpha_l = 1$) [396, 397]. We only fit curves when at least five dose-response points were available, and we only kept curves where the root mean squared error (RMSE, cf. Section 4.3.1) between the actual viabilities and the corresponding points on the fitted curve was ≤ 0.3. This threshold was previously used for data curation of the GDSC database to identify dose-response curves of satisfactory quality [166, 192]. From the fitted curves, we derived two measures of monotherapy drug responses, namely IC50 values and our novel sensitivity measure, the CMax viability, which was introduced in Chapter 7. For the computation of IC50 values, we intersected the dose-response curves with a horizontal line with y-intercept of 0.5 (cf. Figure 3.2). For the computation of CMax viabilities, we evaluated the function of the fitted curve at the drug's CMax concentration (cf. Figure 8.2A).

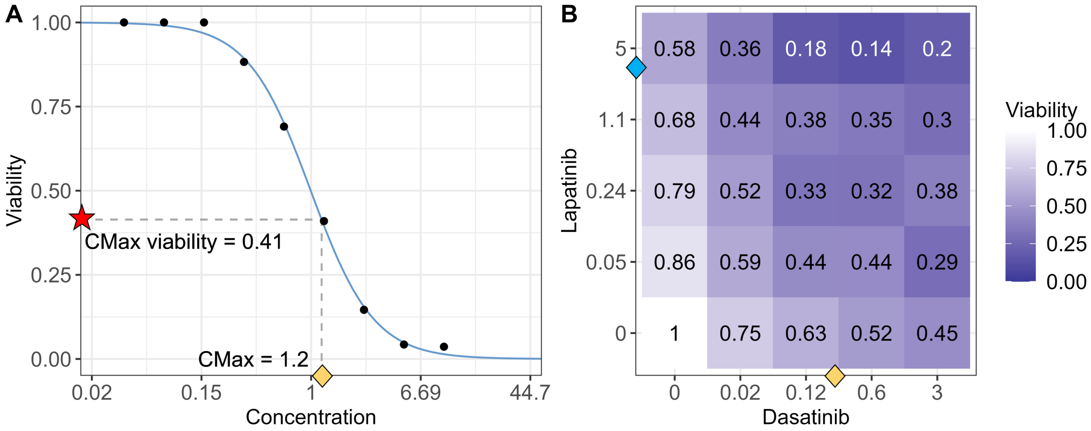


Figure 8.2: Exemplary dose-response curve and matrix. Subfigure A depicts a dose-response curve (blue) for the monotherapy treatment of a cancer cell line (COSMIC ID 683667) with the drug Vorinostat. The fit is based on nine dose-response points (black). The yellow diamond marks the CMax concentration of Vorinostat (1.2μM), and the red star marks the corresponding CMax viability (0.41) derived from the curve. Subfigure B depicts a dose-response matrix for the combination treatment of a cell line (COSMIC ID 909755) with the drugs Dasatinib and Lapatinib. The x- and y-axes denote the respective treatment concentrations. The yellow and blue diamonds approximately mark the CMax concentration of both drugs, which are used to limit the considered concentration combinations for computing the combination CMax viability.

For combination therapies, we developed a variation of the CMax viability that we call the *combination CMax viability*. It can be derived from an actual/predicted dose-response matrix (cf. Figure 8.2B). Our initial idea was to interpolate the values in the dose-response matrix to derive the relative viability when administering the CMax concentration of both combination drugs simultaneously. However, two synergistic drugs may have certain concentration windows with particularly high synergy/effectiveness [390]. Thus, the smallest viability may be reached at a concentration combination smaller than the CMax concentrations. (Note that this should not happen for the dose-response curves we used to compute the CMax viability for monotherapies since these curves are monotonically decreasing.) Consequently, we considered the entire concentration range below the respective CMax values to compute our sensitivity measure: Conceptually, we want to derive the smallest viability within the area defined by the two concentration windows of the drugs limited by their respective CMax concentrations. To this end, we linearly divided the concentration intervals from 0 to the CMax for each drug into 100 equally spaced concentrations, resulting in $100 \cdot 100 = 10,000$ concentration combinations. For each combination, we estimated its relative viability through bilinear interpolation from the full dose-response matrix. Finally, we define the minimum of all 10,000 values as the combination CMax viability.

As the CMax denotes the maximal feasible treatment concentration for a drug monotherapy, it may not be feasible to administer the CMax concentration of two drugs in combination. Yet, we believe that the respective CMax concentrations are a reasonable upper bound for the computation of combination CMax viabilities. Note also that administering the CMax concentration for monotherapies might likewise not be feasible in all cases. Furthermore, the presented approach can theoretically be applied to any desired concentrations other than CMax.

8.2.6 Implementation

MACCS fingerprints and physico-chemical drug properties were calculated using RDKit v.2023.3.2 [394] based on the SMILES drug representations provided by DrugComb. Dose-response curves were calculated using the *drc* R-package v.3.0-1 [396]. Sensitivity measures, namely the IC50 and (combination) CMax viability, were derived from the drug response data using custom R scripts. In particular, we used the *pracma* package v.2.4.2 [398] to perform bilinear interpolation for the calculation of the combination CMax viability.
All prediction models were implemented in Python 3.11: Random forests and elastic net models were implemented using *scikit-learn* v.1.5.0 [360], while neural networks were implemented using *tensorflow* v.2.16.1 [330] with GPU support. The hyperparameters for each algorithm are provided in Table F.2. Our implementations are available at `https://github.com/unisb-bioinf/Drug_Combination_Sensitivity_Prediction`.

8.3 Results

In the following, we present the results of applying our novel models for predicting drug combination sensitivity to data from the DrugComb database. We assess the model performance for different inputs and ML algorithms. Additionally, we investigate the feasibility of reconstructing drug sensitivity measures from the model predictions and performing treatment prioritization.

8.3.1 Overall Performance Comparison

We trained models that predict the relative inhibition of cell growth after a (combination) treatment using three ML algorithms: random forests, neural networks, and elastic nets. To represent drugs in the model input, we investigated four settings (OneHot, OneHotTar, MACCS, PhysChem), which are shown in Figure 8.1 and described in Section 8.2.1.2. The optimized hyperparameters for each model are provided in Appendix Table F.3.

Figure 8.3 shows the performance of all investigated models in terms of test MAE. The first row depicts the results for the entire test data, while the second and third row focus on the data subsets representing mono- and combination therapies, respectively. Across all four settings, random forests had the lowest error, followed by neural networks, while elastic net had the worst performance. An exception is the PhysChem setting, where neural networks were outperformed by elastic nets.
The smallest test error (MAE of 12.14) was achieved using a random forest with MACCS fingerprints as input. Additionally, even the worst-performing random forest model (OneHot, MAE of 13.04) still outperforms the best neural network (OneHot, MAE of 14.08) and elastic net (OneHotTar, MAE of 16.46) models. Thus, the choice of ML algorithm seems to have a stronger impact on performance than the choice of input features, even though the different input representations differ considerably. Notably, adding drug targets slightly improves predictions for random forest and elastic net but has the opposite effect for neural networks. Overall, our findings here match our observations in Chapter 5, where random forests also outperformed neural networks for sensitivity prediction. However, while elastic nets likewise outperformed neural networks in Chapter 5, this is not the case here.

To further contextualize the obtained errors, we compare them to two baseline models: A simple baseline model that always predicts the mean inhibition of the training data has a test MAE of 24.2. Consequently, our best model (MACCS random forest) improves this baseline by 50%. Next, we consider a more advanced baseline model that distinguishes different drugs and drug combinations but does not take the treatment concentration(s) into account: For monotherapies, this model predicts a drug-specific baseline as the mean inhibition of all monotherapy entries from the training data that contain the respective drug. This means we average the observed responses of this drug over the different cell lines and treatment concentrations. Analogously, for combination therapies, the baseline model predicts a combination-specific baseline as the mean inhibition across all training entries containing the respective drug combination. This more advanced baseline model has a test MSE of 19.74, which our best model still improves by 37%. While all random forest models outperform this baseline, some elastic nets and neural networks perform worse.

When investigating mono- and combination therapies separately (cf. Row 2 and 3 of Figure 8.3), the same overall trends can be observed, with the random forest MACCS model again having the smallest error. Generally, both types of therapies can be predicted similarly well, even though the training data contains slightly more combination (60%) than monotherapy data (40%).

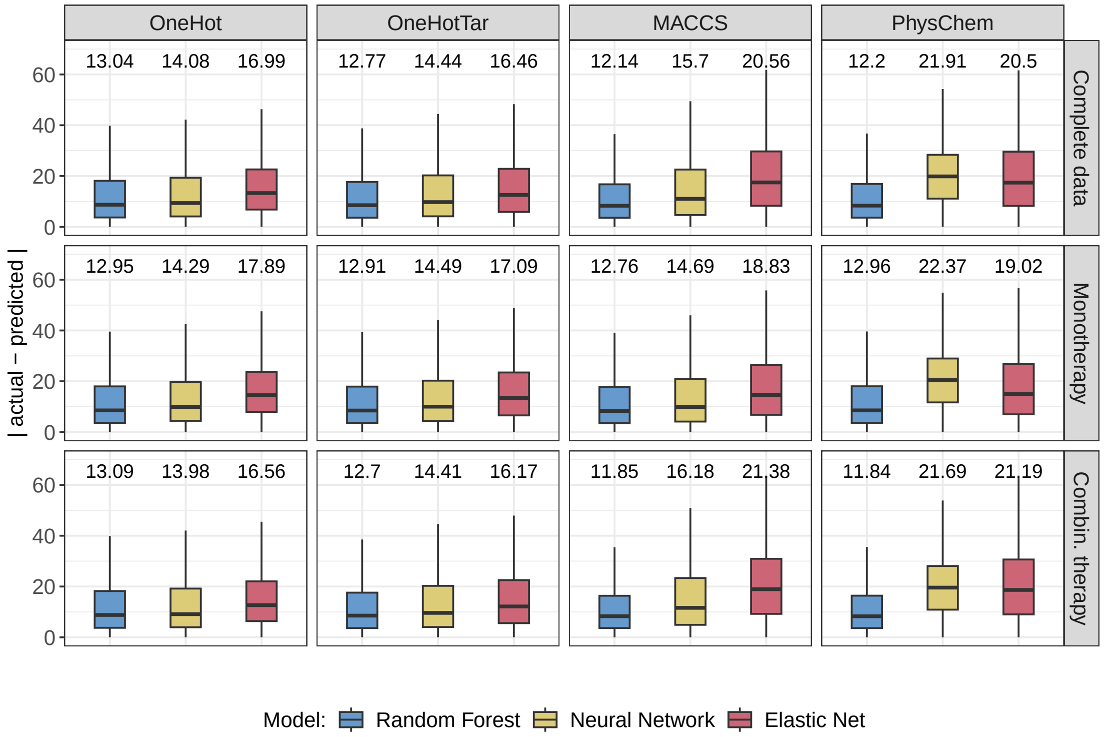


FIGURE 8.3: Test set performance. This figure shows the prediction errors (here: the absolute difference between actual and predicted values) for each investigated setting (columns) and ML algorithm (coloring). The first row shows the results for the entire test dataset, while the second and third row show the results for the data subsets corresponding to mono- and combination therapies, respectively. On top of each boxplot, the mean absolute error (MAE) is shown.

Besides the MAE, we also investigated the Pearson correlation coefficient (PCC) between the actual and predicted inhibitions. The overall PCC for the best-performing model was 0.8 (0.77 for monotherapies, 0.82 for combination therapies). However, computing correlations across the entire data inflates the PCC: Since some drugs/combinations generally have lower/higher inhibitions than others, even our baseline model (as introduced above) that always predicts the mean response for each drug/combination achieves a relatively large correlation of 0.5 (0.44 for monotherapies, 0.53 for combination therapies). Thus, we additionally computed the mean per-drug PCC of our models for monotherapies (0.58) and the mean per-combination PCC for combination therapies (0.56) (cf. Appendix Figure F.1)[1]. These values have a similar magnitude to what we previously observed for monotherapy sensitivity prediction in Chapter 7.

[1] Our baseline model makes constant predictions for each drug/combination. Thus, we cannot compute a baseline PCC per drug/combination since the standard deviation of the predicted values is 0 (cf. Equation 4.66). However, adding small random noise with mean 0 to the baseline predictions would result in a baseline PCC of 0.

In the introductory section of this chapter, we mentioned two methods by Zheng et al. [46] and Julkunen et al. [389] that also predict concentration-specific drug responses. However, their results are not comparable to ours since we investigate the performance for unknown cell lines, which cannot be evaluated using the other two methods. It is known that the cell-line blind scenario increases errors considerably compared to making predictions for known cell lines [302, 399].

To, nevertheless, assess how our random forest MACCS model would perform for known cell lines, we retrained the model using a random split of the available data into a training (80%) and test set (20%). This split does not guarantee that cell lines in the test set were unseen during model training. Note that we still ensured that duplicated entries denoting the same treatment are either exclusively contained in the training or the test set (cf. Section 8.2.4).

With a PCC of 0.96 and RMSE of 8.41, our performance for known cell lines is comparable to that reported by Zheng et al. (PCC = 0.98, RMSE = 7.12) [46] and Julkunen et al. (PCC = 0.97, RMSE = 9.86 in cross-validation; PCC = 0.92 on validation data) [389]. However, the dataset used in our analyses is much larger and more heterogeneous, comprising 947 cell lines, 265 drugs, and 9,535 drug combinations from different sources. In contrast, Zheng et al. solely used the O'Neil dataset (39 cell lines, 38 drugs, 583 drug combinations) [140], which is known to be of high quality [46, 50], whereas Julkunen et al. solely used the AstraZeneca DREAM dataset (85 cell lines, 118 drugs, 910 drug combinations) [194].

8.3.2 Range Performance Comparison

Next, we investigated whether certain inhibition ranges can be predicted more accurately than others. Data points with high inhibition represent cases where the treatment greatly reduced the amount of viable cells. As discussed in Chapter 6, such data points are commonly underrepresented in drug screening datasets [36–38]. They are, however, of particular interest for personalized therapy, where the most effective treatment options for a given patient should be determined.

Figure 8.4 shows the distribution of test MAEs for different inhibition intervals in the range $(-25, 100]$. This range covers 99% of the training and test data. Predictions are (on average) most accurate in the interval $(0, 25]$ followed by the interval $(-25, 0]$. As the actual inhibition increases, the error increases as well. Consequently, the most effective treatments are predicted the least accurately. As mentioned above, this could be explained by the amount of available training data for each interval: Most data is located in the intervals $(0, 25]$ (41%) and $(-25, 0]$ (25%), while each of the other intervals contains only around 10% of the data.

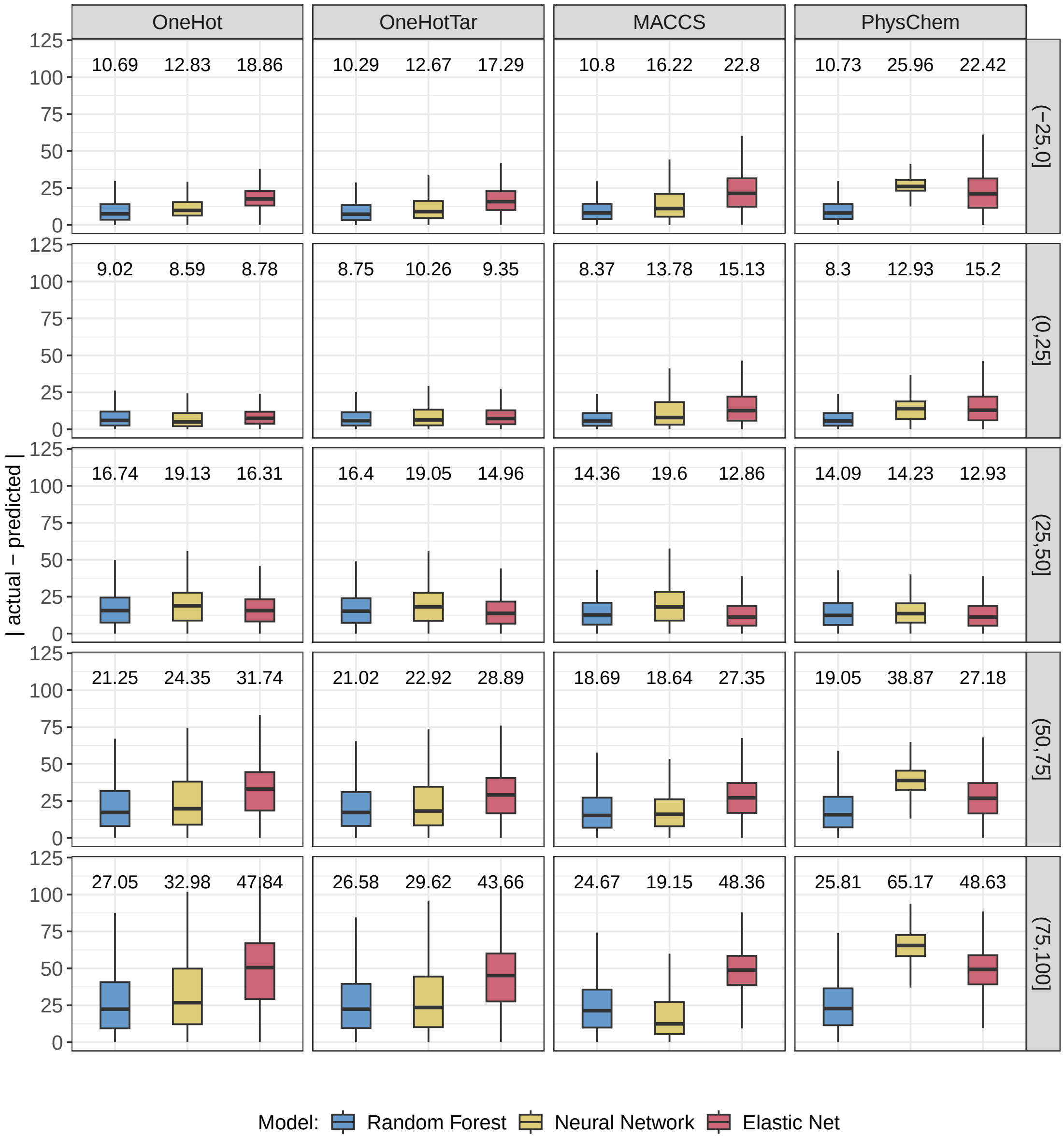


Figure 8.4: Test set performance for different inhibition ranges. This figure shows the prediction errors (in terms of the absolute difference between actual and predicted values) for each setting (columns) and each investigated ML algorithm (coloring). Each row shows the performance for a different interval of actual relative inhibitions. On top of each boxplot, the MAE is shown.

To increase the importance of the underrepresented intervals in model training, sample-specific weights can be employed (cf. Section 4.2). In Chapter 6, we showed that such weights can notably improve predictions for drug-sensitive samples. Thus, we also tried to incorporate sample weights into our models presented here. Unfortunately, the weights had little impact on predictions, especially for the cases with the highest inhibition (see Appendix Figure F.2).

8.3.3 Correlation of Duplicated Entries

As discussed in Section 8.2.2, for the MACCS and PhysChem settings, the same treatment can be described by two different input representations by switching the order of the considered drugs in the input features (cf. Figure 8.1). Hence, we decided to include both input representations in the training and test data of our models. Ideally, predictions for both input representations should correlate well. Figure 8.5A shows the correlation of the duplicated predictions for the random forest MACCS model. As desired, predictions are highly correlated (PCC $\approx$ 1) and the mean absolute difference between them is very small (0.8). Figure 8.5B shows the same analysis for a model where we removed the duplicated entries from the training data. Even though the correlation is still high (PCC = 0.82), it decreased strongly, while prediction differences increased notably to 9.12 on average. The mean PCCs per drug (for monotherapies) and per drug combination are 0.98 and 0.97 for the duplicated training data and decrease to 0.78 and 0.86 for the non-duplicated training data. This is also represented in the test error, where the model with duplicated training entries achieved an MAE of 12.14 compared to 14.6 for non-duplicated entries. Similar trends can also be observed for the PhysChem setting (see Appendix Figure F.3).

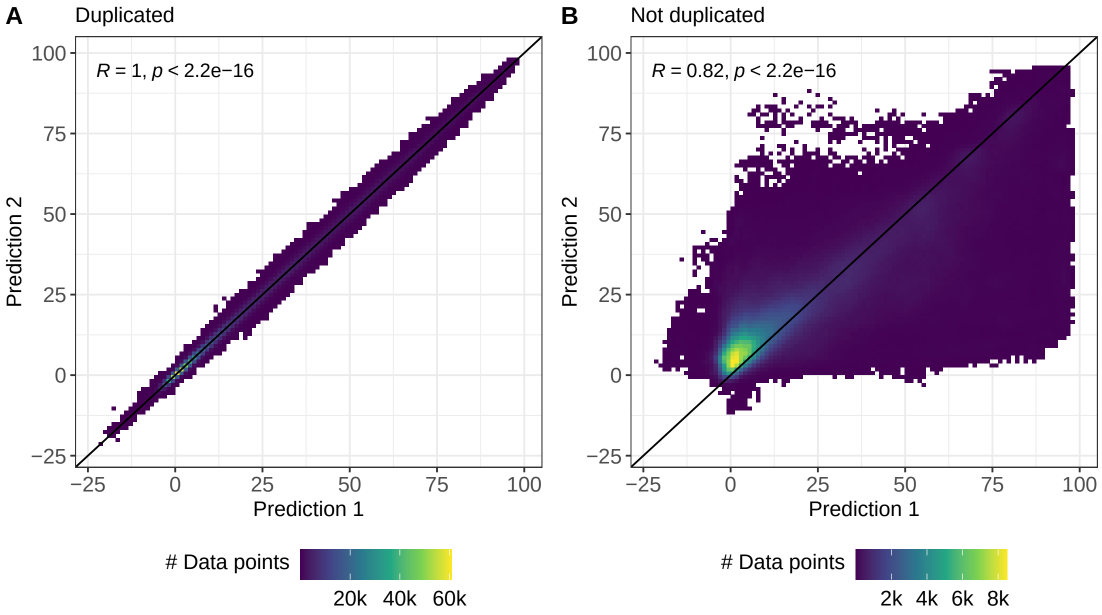


FIGURE 8.5: Correlation of duplicated entries from the test data. This figure shows the correlation between the model predictions for duplicated entries. Duplicated entries refer to the same treatment (i.e., the same drug-(drug-)cell combination and same drug concentration(s)) which can be represented by two different model inputs through swapping the features of the respective drugs (cf. Figure 8.1). Subfigure A shows the test predictions when including duplicated entries in the training data, while Subfigure B shows the predictions when the training data contains only non-duplicated entries. In both figures, the black diagonal line represents the identity, and R denotes the PCC.

8.3.4 Reconstruction of Drug Sensitivity Measures

A benefit of predicting concentration-specific inhibition values is that dose-response curves and matrices can be reconstructed from the model predictions. These curves and matrices can then be used to compute various measures of drug sensitivity or synergy. Since Vlot et al. discourage the computation of arbitrary synergy scores on large-scale data [43], we reconstructed three measures of drug sensitivity using the MACCS random forest: IC50 values, our self-developed measure called CMax viability for monotherapies (cf. Chapter 7), and a modification of this measure for drug combinations, which we call the combination CMax viability (cf. Section 8.2.5). Unlike conventional sensitivity measures like the IC50 or AUC, the (combination) CMax viability is comparable across drugs and drug combinations (see Section 3.2.4.1 for a thorough discussion of the across-drug comparability of sensitivity measures). Consequently, it can be used to prioritize drugs or drug combinations (i.e., rank them by their effectiveness) for a given cell line, which will be investigated in the next section.

In total, we could compute both the actual and predicted CMax viabilities for 7,352 out of 32,564 cell line-drug combinations. The decreased number of combinations stems from the fact that CMax concentrations were only available for 77 of the 265 investigated drugs. Figure 8.6 depicts the prediction errors for the reconstructed monotherapy CMax viability values. The mean MAE averaged over all drugs is 0.12 and the mean MSE is 0.04, which is comparable to the error we achieved in Chapter 7 when predicting CMax viabilities directly using SAURON-RF (MSE = 0.03) or a slightly adjusted version of DeepDR by Chiu et al. [292] (MSE = 0.09).

To obtain a baseline error, we use a baseline model analogous to the baseline introduced in Section 8.3.1. For each drug/combination, the baseline model predicts the drug's/combination's mean viability observed in the training data. The mean is computed by averaging the viability of all mono-/combination therapy entries (involving different cell lines and treatment concentrations) of the respective drug/combination. This baseline model makes constant predictions for a given drug/combination at every treatment concentration. Consequently, the baseline prediction for the CMax concentration, i.e., the baseline CMax viability, is also equal to this constant. This results in a baseline MAE of 0.2, which our model improves by 40%. The overall PCC is 0.58 for the CMax viabilities and 0.41 for the baseline. However, the drug-specific PCC is only 0.1 (cf. Figure 8.6B).[2]

[2] While a drug-specific baseline PCC cannot be computed for constant predictions (where the standard deviation is 0), adding small random noise with mean 0 to these constant predictions results in a baseline PCC of 0.

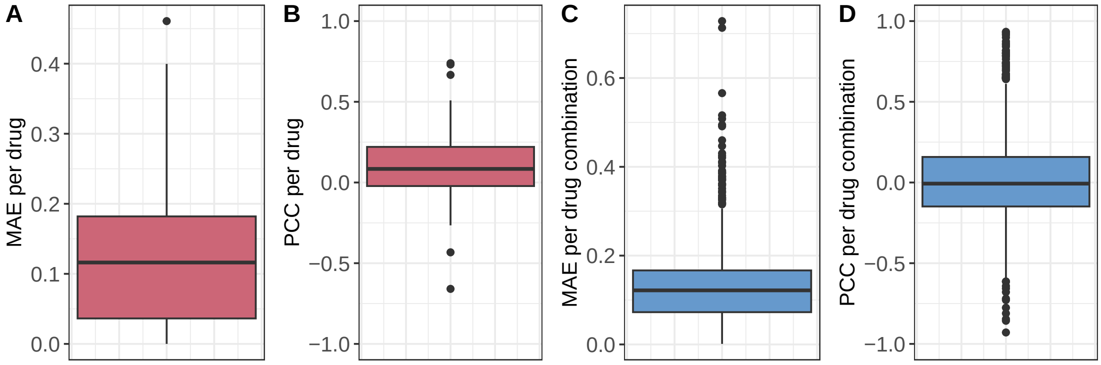


FIGURE 8.6: Reconstruction of (combination) CMax viabilities from predicted dose-response curves/matrices. Subfigures A and B (red) show the distribution of MAE and PCC per drug for reconstructing CMax viabilities using dose-response curves fit on the test set monotherapy data. Subfigures C and D (blue) show the distribution of MAE and PCC per drug combination for reconstructing combination CMax viabilities using dose-response matrices derived from the test set drug combination data.

To investigate the reasons for these low drug-specific correlations, we developed three hypotheses:

Hypothesis 1: The reconstructed dose-response curves might not accurately model the CMax viability in cases where concentrations exceeding the CMax concentration of the respective drug have not been screened.

Hypothesis 2: Even if sufficiently large concentrations were screened, the increased prediction errors for data points with high inhibition (cf. Figure 8.4) might make the curve-fitting unreliable in areas of high inhibition, affecting the derived measures.

Hypothesis 3: The two-step process of first reconstructing a curve and then deriving the CMax viability from the curve is inferior to directly predicting the CMax viability with our models.

A detailed evaluation of these hypotheses can be found in Appendix F.1. Unfortunately, none of our evaluations provided a conclusive explanation for the low correlations. Thus, we conclude that even though prediction errors (MAE) are relatively small and comparable to our previous work, the derived measures cannot be used to compare the effect of a drug monotherapy on different cell lines. We obtained similar results for the combination CMax viability and the IC50, which are depicted in Figure 8.6C and D and Appendix Figure F.6.

Nevertheless, we would like to highlight that such an evaluation of drug-specific correlations as conducted here is frequently not performed for drug sensitivity and synergy prediction: In Table 8.1, we compare the investigated settings and analyses for 55 state-of-the-art methods.

Table 8.1: Drug sensitivity/synergy prediction literature. This table lists 55 approaches for drug sensitivity and synergy prediction. For each approach, it is denoted whether dose-specific predictions can be made, whether a cell-blind evaluation was performed, and whether drug-/combination-specific correlations are provided (Pearson correlation for regression, Matthews correlation for classification). Rows marked in light blue denote approaches working with drug combination data for sensitivity or synergy prediction. Rows marked in yellow denote multi-drug models for monotherapy sensitivity prediction. Finally, rows without coloring denote single-drug models for monotherapy sensitivity prediction. Table continues on next page.

Model	**Dose-specific predictions**	**Cell-blind**	**Correlation per drug or combin.**
Rahman & Pal (2016) [387]	✓	✓	✓
LOBICO (2016) [36]	✗	✓	✗
HARF (2017) [285]	✗	✓	✗
RWEN (2018) [38]	✗	✓	✗
CDRscan (2018) [289]	✗	✗	✓
QRF (2018) [290]	✗	✓	✓
NCFGER (2018) [291]	✗	✗	✓
Deep-Resp-Forest (2019) [179]	✗	✓	✗
netBITE (2019) [293]	✗	✓	✓
FRF (2019) [388]	✓	✓	✗
Deng et al. (2020) [294]	✗	(✓)[†]	✗
Ahmed et al. (2020) [295]	✗	✓	✓
MERIDA (2021) [37]	✗	✓	✗
SAURON-RF (2022) [39]	✗	✓	(✓)[§]
reliable SAURON-RF (2023) [40]	✗	✓	✓
GADRP (2023) [323]	✗	✗	✗
Menden et al. (2013) [282]	✗	✗	✗
Zhang et al. (2015) [283]	✗	✗	✓
SRMF (2017) [284]	✗	✗	✓
Stanfield et al. (2017) [177]	✗	(✓)[†]	✗
HNMDRP (2018) [307]	✗	✗	✗
MOLI (2019) [314]	✗	(✓)[†]	✗
DeepDR (2019) [292]	✗	✓	✗
MinDrug (2021) [381]	✗	✓	✗
PathDSP (2021) [297]	✗	✓	(✓)[‖]
ADRML (2020) [296]	✗	✗	✓
GraphDRP (2021) [299]	✗	✓	✗
RAMP (2022) [315]	✗	(✓)[†]	✗
NeRD (2022) [322]	✗	✓	✗
Precily (2022) [300]	✗	✓	✓
KBMTL (2014) [301]	✗	✓	✗
DeepCDR (2020) [303]	✗	✓	(✓)[‖]

Continuation of Table 8.1.

Model	Dose-specific predictions	Cell-blind	Correlation per drug or combin.
Pivetta et al. (2013) [400]	✓	✗	✗
Gu et al. (2015) [401]	✓	(✗)‡	(✓)‡
DIGRE (2015) [402]	(✓)*	(✗)‡	✗
Zimmer et al. (2016) [403]	✓	✗	✓
Hsu et al. (2016) [404]	✗	(✓) no ML	✗
Zimmer et al. (2017) [405]	✓	✗	✗
SyDRa (2017) [306]	(✓)*	(✗)‡	✗
Jeon et al. (2017) [406]	✗	✗	✗
TreeCombo (2018) [287]	✗	✗	✗
TAIJI (2018) [407]	✗	not stated	✗
QPOP (2018) [408]	✓	(✗)‡	✗
Xia et al. (2018) [409]	✗	✗	✗
Sidorov et al. (2019)[410]	✗	✗	✗
DECREASE (2019) [390]	✓	✗	✗
Ling & Huang (2020) [411]	✓	✗	✓
Julkunen et al. (2020) [389]	✓	✗	✗
REFINED CNN (2021) [298]	✗	✗	✗
Zheng et al. (2021) [46]	✓	✗	✗
Correia et al. (2021) [412]	✓	(✗)‡	✗
Pinoli et al. (2022) [413]	✗	✓	✗
DeepSynergy (2018) [302]	✗	✗	✓
Kim et al. (2021) [304]	✗	✗	✗
MatchMaker (2022) [305]	✗	✗	✓

* Post-treatment gene expression for dose-specific monotherapies is required for DIGRE [402] and SyDRa [306] such that predictable doses are heavily constrained by data availability.

† These models use different data sources for training and testing, e.g., the GDSC and CCLE databases. However, since there is considerable overlap in the screened cell lines of large-scale drug-screening panels [175, 414], the same cell lines may be contained in both datasets. An exception are MOLI [314] and RAMP [315], where the models are additionally evaluated using xenograft and patient data, respectively. Deng et al. [294] additionally perform a leave-one-out cross-validation on the cell lines but do not provide cell-blind evaluations for a dedicated test set.

‡ These approaches are only trained and evaluated using ≤ 2 cell lines.

§ SAURON-RF [39] only provides drug-specific correlations for classification but not for regression.

|| The drug-specific correlation evaluations in PathDSP [297] and DeepCDR [303] were not performed for the cell-blind setting.

Most approaches do not assess drug-specific correlations, especially not for the cell-blind and multi-drug setting. Thus, similar problems may often go undetected. Due to the novelty of our prediction approach, there is no method we could directly compare our findings to. However, there exist some related approaches that may aid in contextualizing our results: Firstly, our analyses presented earlier show that our models are competitive in performance to those by Zheng et al. [46] and Julkunen et al. [389] when making predictions for known cell lines. Note that both approaches do not provide drug- or combination-specific correlations.

For cell-blind evaluations on monotherapy data, we found three related approaches that provide drug-specific correlations: As discussed in Chapter 7, our method SAURON-RF achieves a mean PCC of 0.56 when directly predicting CMax viabilities using drug-specific models [40]. In the same chapter, we also show that an adjusted version of the multi-drug model DeepDR by Chiu et al. [292] achieves a PCC of 0 for the same task. In comparison, Precily by Chawla et al. uses multi-drug models for predicting IC50 values and achieves mean PCCs between ca. 0.18 and 0.5 for different ML algorithms [300]. Lastly, Rahman and Pal achieve mean PCCs between 0.29 and 0.44 when reconstructing AUC values from predicted dose-response curves [387]. While not directly comparable to our approach, these works underline that at least weak to moderate drug-specific correlations can be achieved (1) for predicting CMax viabilities, (2) when using multi-drug models, or (3) when deriving sensitivity measures from predicted curves. Yet, it remains to be investigated further if and how comparable results could be achieved when combining all three factors and also considering combination therapies, thereby enabling predictions for arbitrary drugs/combinations and measures, which we aim to achieve here.

8.3.5 Treatment Prioritization

In our final analysis, we investigate how accurately drugs and drug combinations can be prioritized for a given cell line based on the predictions of the MACCS random forest: For each cell line in the test set, we used the computed CMax viabilities for the monotherapy and combination data to achieve a ranking of drugs and drug combinations from most to least effective, i.e., from the smallest to largest CMax viability. Drug prioritization is supposed to mimic a personalized treatment scenario with the goal of generating a list of the most effective treatment suggestions for a given patient. The results are shown in Figure 8.7, where the first row shows the results for monotherapies only, while the second row shows the results when combining mono- and combination therapies into one list.

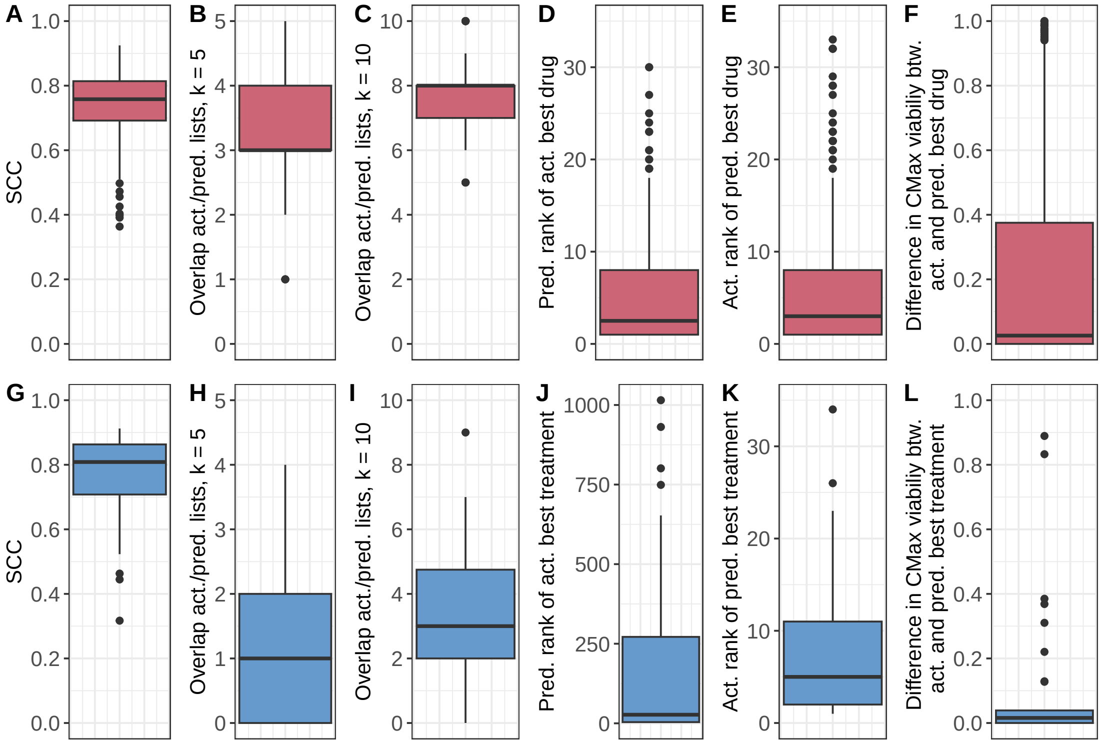


FIGURE 8.7: Treatment prioritization. This figure depicts the test set prioritization results for mono- and combination therapies. Subfigures A to F (red) focus on the prioritization of monotherapies, including: (A) the SCC between the actual and predicted rankings for each cell line, (B)/(C) the intersection size between the 5/10 actual and predicted most effective treatments, (D) the predicted rank of the actual most effective treatment, (E) the actual rank of the treatment predicted to be most effective, and (F) the difference between the actual CMax viabilities for the actual and predicted most effective treatment. Subfigures G to L (blue) show the analogous prioritization results when combining mono- and combination treatments into one list.

For monotherapies, the Spearman correlation coefficient (SCC) between the actual and predicted rankings was 0.74 (baseline as defined in the previous section: 0.54). While an accurate ranking for the entire drug list is desirable, one would typically place more emphasis on the correct identification of the most effective treatments. Thus, we computed the mean overlap between the first k elements of the actual and predicted rankings. For monotherapies, the average length of the predicted drug lists is 31.15. The average overlap between the top $k = 5$ and $k = 10$ actual and predicted most effective drugs is 3.16 (baseline: 2.14) and 7.68 (baseline: 6.55), respectively (results for further k are shown in Appendix Figure F.4). This shows that our rankings successfully identify most of the best treatment options. Furthermore, the median rank of the actually most effective drug in the predicted ranking is 2.5 (baseline: 8), and the median rank of the drug predicted to be most effective in the actual list is 3 (baseline 6). Additionally, the median difference between the true CMax viabilities of the actual most effective and predicted most effective drugs is only 0.02 (baseline 0.31). Consequently, the first drugs in our predicted lists are highly similar in effectiveness to the actually most effective option.

The second row of Figure 8.7 shows the analogous prioritization results when combining mono- and combination treatments into one list. The SCC of 0.76 (baseline: 0.62) is comparable to the results for monotherapies. Since the average list length is much greater when including drug combinations (838.62), the overlaps at $k = 5$ (1.26, baseline: 0.68) and $k = 10$ (3.38, baseline: 2.09) are lower (cf. also Appendix Figure F.5). Furthermore, the median rank of the actually best treatment in the predicted list (27, baseline: 170.5) and of the predicted best treatment in the actual list (9.5, baseline: 12) decrease. Still, results clearly improve over the baseline. Furthermore, the median difference in viability between the actually most effective treatment and the treatment predicted to be most effective remains small (0.02, baseline 0.03).

8.4 Discussion

Administering not only single but multiple drugs in combination is common in cancer treatment. Such combination therapies can enhance treatment efficacy and reduce the risk of developing treatment resistance [41]. However, while drug response datasets for monotherapy data have been available for more than a decade, large-scale datasets for combination therapy have only become publicly available more recently, e.g., the DrugComb database [46, 50]. While the DrugComb data have extensively been studied for predicting drug synergy, they are still underused for predicting drug sensitivity, especially with the focus on making personalized treatment recommendations. For this application, we found the scores that are widely used for synergy prediction less suited due to various reasons discussed in this chapter.

To exploit the available drug combination data for predicting drug responses without relying on synergy scores, we investigated several ML algorithms and model architectures that directly predict concentration-specific drug responses in the form of relative inhibitions. This approach has various benefits for personalized treatment recommendation: First, our approach allows the reconstruction of dose-response curves and matrices from the model predictions. From these curves/matrices, various sensitivity or synergy measures can be derived. Inspecting individual curves/matrices can also aid in validating the underlying assumptions for certain measures. Next, our approach can predict both mono- and combination therapies. Additionally, our approach allows for making predictions for unseen cell lines. This *cell-blind* scenario mimics the important but often overlooked application of assessing drug responses for a new patient [313]. Together with our novel sensitivity measure, the *(combination) CMax viability*, this framework finally enables the prioritization of both mono- and combination therapy options for unseen cell lines (patients).

Our evaluations on the DrugComb database show that our models substantially improve baseline models and show very little variation when predicting the same treatment using different input representations. Moreover, our models are competitive with state-of-the-art approaches when making predictions for known cell lines. Furthermore, we achieved strong correlations for treatment prioritization that, likewise, improve over the respective baseline models.

Our use of baseline models was motivated by the fact that there is no existing ML approach we could directly compare our models to in the cell-blind scenario. However, our baseline models do not account for the fact that an increase in concentration generally leads to an increase in relative inhibition. Thus, more complex baselines might provide a more precise assessment of our models' capabilities and potential shortcomings.

While our analyses demonstrate the strengths of our approach, they also reveal weaknesses of directly predicting relative inhibitions: We observed increased prediction errors for samples with high inhibitions, corresponding to cases of treatment sensitivity. As discussed extensively in Chapter 6, this issue is relatively well-known for classification but has rarely been discussed or addressed for regression [38, 39]. Additionally, when reconstructing drug sensitivity measures (IC50 and CMax viability), the drug-specific correlations between the actual and predicted values are weak.

Three main factors that could be adjusted to address such challenges are (1) the choice of ML algorithm, (2) the choice and representation of input features, and (3) the used dataset:

1. ML algorithm:

We investigated three ML algorithms, namely neural networks, random forests, and elastic nets. Due to the large number of tunable parameters, neural networks generally require relatively large amounts of training data. Since the dataset investigated here is much larger than those in the previous chapters, we expected neural networks to outperform the other approaches. However, in line with our findings in Chapter 5, random forests were superior in all investigated settings. Likewise, several studies found that deep learning does not improve over conventional (tree-based) algorithms for making predictions on tabular data [316–318]. Thus, even though neural networks are highly popular for sensitivity and synergy prediction, tree-based methods seem promising for predicting the response to combination therapies.

In general, a plethora of further (potentially more sophisticated) ML approaches could be used to predict relative inhibitions. However, as discussed in Chapter 5 and also by Li et al. [24], more complex approaches are not necessarily superior to simpler ML algorithms, and careful evaluation is required to ensure a fair performance comparison.

2. Input features and representation:
For the characterization of cell lines in the model input, several sources found gene expression to be the most informative omics type for predicting drug responses [23, 176, 292]. However, including further omics or a priori knowledge, e.g., known sensitivity biomarkers or protein interactions, might improve predictions.
Similarly, further drug properties, e.g., Morgan fingerprints [415] could be investigated, or graph neural networks could be employed to represent drugs as molecular graphs. However, the superiority of molecular graphs over fingerprints for sensitivity/synergy prediction and drug discovery has been questioned [34, 416].
Additionally, although some of our input representations enable predicting responses for previously unseen drugs or for drug combinations consisting of more than two drugs, we did not evaluate these scenarios. While three-drug combination therapies have already been approved for cancer treatment [108], DrugComb does not provide such data.

3. Dataset:
Given the size of the DrugComb database used to train our models, hardware restrictions become a limiting factor for ML. Despite training models on a compute cluster with machines of 500 gigabytes of working memory, we had to limit the number of drugs and features considered in our analyses (cf. Sections 8.2.1 and 8.2.2). Nevertheless, with 947 cell lines, 265 drugs, and 9,535 drug combinations, the dataset investigated here remains notably larger compared to other approaches working on drug combination data [46, 183, 287, 302, 306, 389, 410].
Generally, a large amount of training data benefits model training and robustness. Especially our neural networks could benefit from a larger training dataset since they have a great number of tunable parameters compared to the random forest and elastic net models. However, if the dataset is heterogeneous, e.g., due to different data sources as in DrugComb, this may decrease performance compared to models built and evaluated on a more homogeneous dataset. Even though Zagidullin et al. [50] and Liu et al. [417] found the reproducibility between replicates from different datasets in DrugComb satisfactory, disagreement between drug response data from different sources is a well-known problem [23, 175, 418]. Nevertheless, especially for clinical applications, combining data from different sources (e.g., different hospitals) is essential, and models should be able to cope with this degree of heterogeneity. To this end, meta- or transfer-learning methods could be leveraged [419].

Lastly, as discussed in Chapter 4.3, performance alone should not be regarded as the sole building block of a trustworthy model: To assess the reliability of individual predictions, uncertainty estimation frameworks like conformal prediction could be applied as shown in Chapter 7 [40, 267, 272, 377]. Additionally, incorporating interpretability mechanisms [26, 287, 297] into the model design and evaluation can aid in identifying drug or cell line properties that impact the predicted response. This could not only make predictions more comprehensible but also be useful to infer novel mechanisms of drug sensitivity or synergy.

Chapter 9

Predicting Muscle Invasion of Bladder Cancer

The previous chapters focused on the development of accurate and reliable machine learning (ML) models for drug sensitivity prediction. Such ML models should eventually be used in clinical decision support systems to derive personalized treatment strategies for cancer patients. However, drug sensitivity prediction is not the only field in which personalized medicine can benefit from ML. In this chapter, we focus on one cancer type, namely bladder cancer, and investigate how ML can aid in deciding for or against a surgical removal of the bladder to prevent tumor spread.

Bladder cancer (BC) is currently the ninth most common cancer type worldwide [2]. While the average five-year survival rate in the United States is 77% [420], disease outcome is heavily linked to the invasive potential of the tumors. Consequently, one commonly distinguishes between muscle-invasive (MIBC) and non-muscle-invasive (NMIBC) bladder cancer: MIBCs make up around 30% of BCs [421] and are characterized by the tumor growing into the muscle layer of the bladder wall. This muscle invasion is linked to an increased risk of metastases [422] and a notably lower five-year survival rate ($< 39\%$) compared to NMIBCs (70-96%) [421, 423].

For MIBCs, the standard of care is to perform a cystectomy (CYS), where the entire bladder is removed to prevent the spread of the tumor [424]. In contrast, for NMIBCs, a transurethral resection (TUR) is generally sufficient. In a TUR, tumor tissue in the bladder is removed through the urethra while keeping the bladder intact. However, the group of NMIBCs can be further subdivided into low-grade (pTa lg) tumors that are less likely to spread and into more aggressive high-grade (pT1 hg) tumors. While high-grade tumors have not (yet) invaded the muscle layer, they have grown into the lamina propria – a tissue between the bladder lining and the muscle layer – and show potential for muscle invasion and metastasis [425]. Thus, preventive bladder removal may also be advised for these types of tumors.

Currently, the decision on whether a CYS is performed is mainly based on tumor assessment by a surgeon, tumor recurrence, and multifocality, i.e., the presence of multiple individual tumors in the bladder. However, especially for pT1 hg tumors, it remains challenging to characterize their invasive potential, even though some efforts to this end have been made: To estimate tumor progression and patient survival in pT1 hg BC, Van de Putte et al. used a histopathological substaging approach based on the degree of invasion into the lamina propria [425]. In comparison, Dyrskjøt et al. propose a progression score that combines expression values of twelve genes [426]. However, the identified markers are either hard to reproduce or not suited for routine practice.

A more feasible approach for daily practice would be using biomarkers that can be identified using standard laboratory procedures. MiRNAs are one class of such biomarkers that can be assessed not only in tumor tissues but also in bodily fluids such as blood or urine. As discussed in Section 2.1.1.1, miRNAs are small, non-coding RNAs that play a key role in gene regulation. Alterations in miRNA expression have been linked to multiple disease classes, such as cancer and Alzheimer's [427, 428].

In a previous study, Baumgart et al. identified four miRNAs whose expression differed significantly between MIBC and NMIBC tumor samples [429]: miR-138-5p, miR-146b-5p, miR-155-5p, and miR-200a-3p. Furthermore, they found significant MIBC vs. NMIBC expression differences of miR-146b-5p and miR-155-5p in urinary extracellular vesicles (EVs). These EVs are released from healthy and cancerous cells of the urinary tract into the urine and are involved in cell communication and tumor development [430]. Thus, urine samples could be a non-invasive means to assess a patient's risk of developing MIBC.

Based on the findings by Baumgart et al., the aim of the study presented here was to (1) confirm the prognostic potential of the four identified miRNAs, and (2) develop classification models that can accurately distinguish MIBC from NMIBC based on these miRNAs. In total, we investigated data of 369 bladder cancers from two patient cohorts. Our analyses focus predominantly on the distinction between MIBC and pTa lg tumors. While pT1 hg samples are contained in our dataset, follow-up data indicating the potential progression of pT1 hg to MIBC was, unfortunately, only available for a very limited number of samples. Information on disease progression would, however, be crucial to assess whether our models can accurately distinguish between pT1 hg samples that remain non-invasive and those that become invasive over time.

Our results show significant expression differences between MIBC and pTa lg for all four miRNAs in both patient cohorts. The developed classification models can distinguish MIBC from pTa lg with high accuracy, even when only samples obtained via TUR are considered. Additionally, we applied conformal prediction to our models. As discussed extensively in Section 4.3.2.1 and Chapter 7, extending a classification model with conformal prediction yields sets of classes (instead of point-predictions) that are guaranteed to contain the actual class of a sample with a user-defined certainty. Such certainty guarantees are highly desirable for creating trust in the clinical application of ML models. Furthermore, the addition of conformal prediction to our models eliminated all misclassifications in the test cohort. Minimizing erroneous predictions is also crucial for the real-life application of ML, especially in the medical field, where treatment errors may have severe consequences for the patient.

Finally, we applied our models to pT1 hg tumors to investigate their invasive potential. We correctly predicted muscle invasion for five out of six pT1 hg samples with available follow-up data.

Author Contributions

This chapter is based on the following publication:

Eckhart, L.*, Rau, S.*, Eckstein, M., Stahl, P. R., Ayoubian, H., Heinzelbecker, J., ... & Junker, K. (2025) **Machine learning accurately predicts muscle-invasion of bladder cancer based on three miRNAs**. Journal of Cellular and Molecular Medicine, 29(3), e70361
DOI: 10.1111/jcmm.70361
(* authors contributed equally)

Kerstin Junker and Michael Stöckle conceptualized the study. I was involved in processing the data and developed the classification models. These steps were supervised by Kerstin Junker and Hans-Peter Lenhof. I also wrote the manuscript sections describing the training and performance evaluation of the machine learning models, which all authors reviewed.

9.1 Materials and Methods

In the following, we first present the dataset we used for our analyses. Next, we describe the methods we applied to compare miRNA expression between sample groups. Finally, we provide details regarding the trained classification models.

9.1.1 Dataset

Data from 369 bladder cancers from two independent cohorts was analyzed. Cohort 1 comprises 78 samples collected at Saarland University Hospital, and Cohort 2 comprises 291 samples collected at Erlangen University Hospital. Samples for pTa lg and pT1 hg tumors were obtained via transurethral resection (TUR), where tumor tissue in the bladder is removed through the urethra while keeping the bladder intact. MIBC samples were obtained either via TUR or through a cystectomy (CYS), where the bladder is removed entirely. Table 9.1 provides an overview of the distribution of samples in both cohorts.

The expression of the four miRNAs miR-138-5p, miR-146b-5p, miR-155-5p, and miR-200a-3p in the tumor samples was determined using quantitative real-time polymerase chain reaction (qRT-PCR, see Appendix G.1 for details) [431]. Expression was measured as *crossing point* (CP or C_t) values [432]. These values denote the number of PCR amplification cycles until the presence of the respective miRNA was detected in the probe through fluorescence. Consequently, a lower CP value indicates a stronger expression since less amplification was required. CP values for each miRNA were additionally normalized to ΔCP values by subtracting the CP value of miR-361-5p measured in the same experiment. This miRNA is used for normalization as it is thought to have a relatively stable expression in different cancer types [433].

Table 9.1: Dataset overview. This table summarizes the number of investigated tumor samples in each cohort, separated by cancer classification and surgery type. Note that no CYS samples exist for the pTa lg and pT1 hg classes.

Classification	Surgery	Cohort 1	Cohort 2	total
pTa lg	TUR	27	86	113
pT1 hg	TUR	12	108	120
MIBC	TUR + CYS	39	97	136
	TUR	16	23	39
	CYS	23	74	97

9.1.2 Test for Differential miRNA Expression

To test for significant differences in miRNA expression between the different stages of BC, we used the Mann-Whitney U test. The Mann-Whitney U test (also known as the Wilcoxon rank-sum test [434]) is a non-parametric statistical test that evaluates whether two independent random samples are drawn from populations that follow the same distribution [435]. For a two-tailed test, the null hypothesis states that the distributions of both populations are equal, while the alternative hypothesis states that they are not equal.

Consider a set of samples $\{(x_1, y_1), ..., (x_N, y_N)\}$, where each sample i is characterized by a feature value x_i, and a binary class $y_i \in \{c^{\text{pos}}, c^{\text{neg}}\}$. In our application case, each sample is a tumor, x_i denotes the expression value for the miRNA of interest, and c^{pos} and c^{neg} correspond, e.g., to the MIBC and pTa lg classes.

To compute the test statistic, we first convert all values x_i into ranks r_i, where the largest value is assigned rank 1, and the smallest value is assigned rank N [436]. Next, the ranks of all data points belonging to class c^{pos} are added up:

$$R_{pos} = \sum_{i=1}^{N} I(y_i = c^{\text{pos}}) \cdot r_i \tag{9.1}$$

The indicator function I is 1 if sample i belongs to class c^{pos} and 0 otherwise. The Mann-Whitney U-statistic is then computed as follows, where N_{pos} and N_{neg} denote the number of samples with class c^{pos} and c^{neg}, respectively [435]:

$$U = N_{\text{pos}} \cdot N_{\text{neg}} + \frac{N_{\text{pos}} \cdot (N_{\text{pos}} + 1)}{2} - R_{\text{pos}} \tag{9.2}$$

For large sample numbers N, the distribution of U approaches a normal distribution with mean μ_U and standard deviation σ_U [436]. Thus, U can be standardized by computing

$$Z = \frac{U - \hat{\mu}_U}{\hat{\sigma}_U} \tag{9.3}$$

using the following estimates for μ_U and σ_U [436]:

$$\hat{\mu}_U = \frac{N_{\text{pos}} \cdot N_{\text{neg}}}{2} \tag{9.4}$$

$$\hat{\sigma}_U = \sqrt{\frac{N_{\text{pos}} \cdot N_{\text{neg}} \cdot (N_{\text{pos}} + N_{\text{neg}} + 1)}{12}} \tag{9.5}$$

For large sample sizes, a p-value can now be derived from a standard normal distribution as the probability of observing a value that is $\geq |Z|$ or $\leq -|Z|$. For smaller sample sizes, the p-value has to be looked up in a precomputed table [436]. We considered all p-values ≤ 0.05 significant.

9.1.3 ROC Curves for Individual miRNAs

In Section 4.3.1, we described how ROC (*receiver operating characteristic*) curves can be used to assess the performance of a binary classifier based on the predicted class probabilities. Here, we describe a slightly modified approach that we applied to assess how well individual miRNAs can distinguish between MIBC and pTa lg samples without training any ML model.

Consider again a set of tumor samples $\{(x_i, y_i), ..., (x_N, y_N)\}$, where each sample i is characterized by its expression value x_i for the miRNA of interest, and its actual class $y_i \in \{c^{\text{pos}} = \text{MIBC}, c^{\text{neg}} = \text{pTa lg}\}$.

Based on this data, we first determine which of the two classes has the largest median miRNA expression. This class is denoted as $c^{\max}$, while the other class is denoted as $c^{\min}$. Next, we employ a simple classifier based on an expression threshold t. Given t, the class prediction $\hat{y}_i$ for sample i can be derived as [437]:

$$\hat{y}_i = \begin{cases} c^{\max} & \text{if } x_i \geq t \\ c^{\min} & \text{else} \end{cases} \tag{9.6}$$

To obtain the ROC curve, the threshold t is varied, and the corresponding sensitivity and specificity are computed (cf. Section 4.3.1). Finally, the area under the curve (AUC $\in [0, 1]$) can be computed, where larger values indicate a better classification ability (cf. Section 4.3.1).

9.1.4 Classification Models

We trained classification models that distinguish MIBC from pTa lg tumors based on miRNA expression using five machine learning algorithms: boosting trees, k-nearest neighbors, random forests, support vector machines, and vanilla neural networks, i.e., neural networks with only one hidden layer [245]. Detailed descriptions of these algorithms can be found in Section 4.2. Additionally, to determine prediction sets that contain the actual class of a sample with a probability of 90%, we performed conformal prediction using the Summation score described in Section 4.3.2.1 with an error rate of $\alpha = 0.1$.

Two-thirds of samples from the larger cohort (Cohort 2) were randomly chosen for model training. The remaining third of samples was used as calibration data for conformal prediction. Finally, the smaller cohort (Cohort 1) was used for testing.

Additionally, a 5-fold cross-validation was performed on the training data to find the hyperparameters that maximize the area under the ROC curve for each model (cf. Section 4.3.1). The hyperparameters are provided in Appendix Table G.1.

9.1.5 Implementation

All ML models were trained in R using the following packages: *ada* (v.2.0.5) for boosting trees [438], *class* (v.7.3.19) for k-nearest neighbor models [345], *randomForest* (v.4.7.1.1) for random forests [361], *kernlab* (v.0.9-31) for support vector machines [439], and *nnet* (v.7.3.16) for neural networks [345]. The respective model hyperparameters are listed in Appendix Table G.1. Mann-Whitney U tests and ROC curves were computed using GraphPad Prism [440].

9.2 Results

In the following sections, we first investigate how well individual miRNAs can distinguish between MIBC and pTa lg cancers within each cohort using statistical tests and ROC curves. Next, we present the results of our MIBC vs. pTa lg classification models trained on different combinations of the four miRNAs. We then investigate how the models classify pT1 hg samples and whether our predictions align with the disease progression derived from follow-up data. Finally, we discuss the role of the four miRNAs in bladder cancer and whether our findings align with previous research.

9.2.1 Differential miRNA Expression in MIBC vs. pTa lg Tumors

Figures 9.1A and B depict the expression of all four miRNAs in MIBC compared to pTa lg samples for Cohort 1 and 2, respectively. The significance of expression differences between the two groups was assessed using Mann-Whitney U tests. In both cohorts, miR-138-5p and miR-200a-3p were significantly downregulated in MIBC, while miR-146b-5p and miR-155-5p were significantly upregulated, which matches the findings by Baumgart et al. [429].

Next, we split the MIBC group into samples obtained via TUR and CYS and compared both subgroups separately to pTa lg. For CYS-MIBC samples, the expression of all four miRNAs differed significantly from pTa lg in both cohorts. For TUR-MIBC, only miR-146b-5p and miR-155-5p showed significant expression differences. Lastly, when comparing TUR-MIBC and CYS-MIBC samples, miR-146b-5p and miR-200a-3p expression differed significantly in Cohort 2, but no other significant differences were found. For all analyses where TUR and CYS samples were considered separately, the corresponding box plots are provided in Appendix Figures G.1 to G.4. Furthermore, p-values for all analyses are provided in Appendix Table G.3.

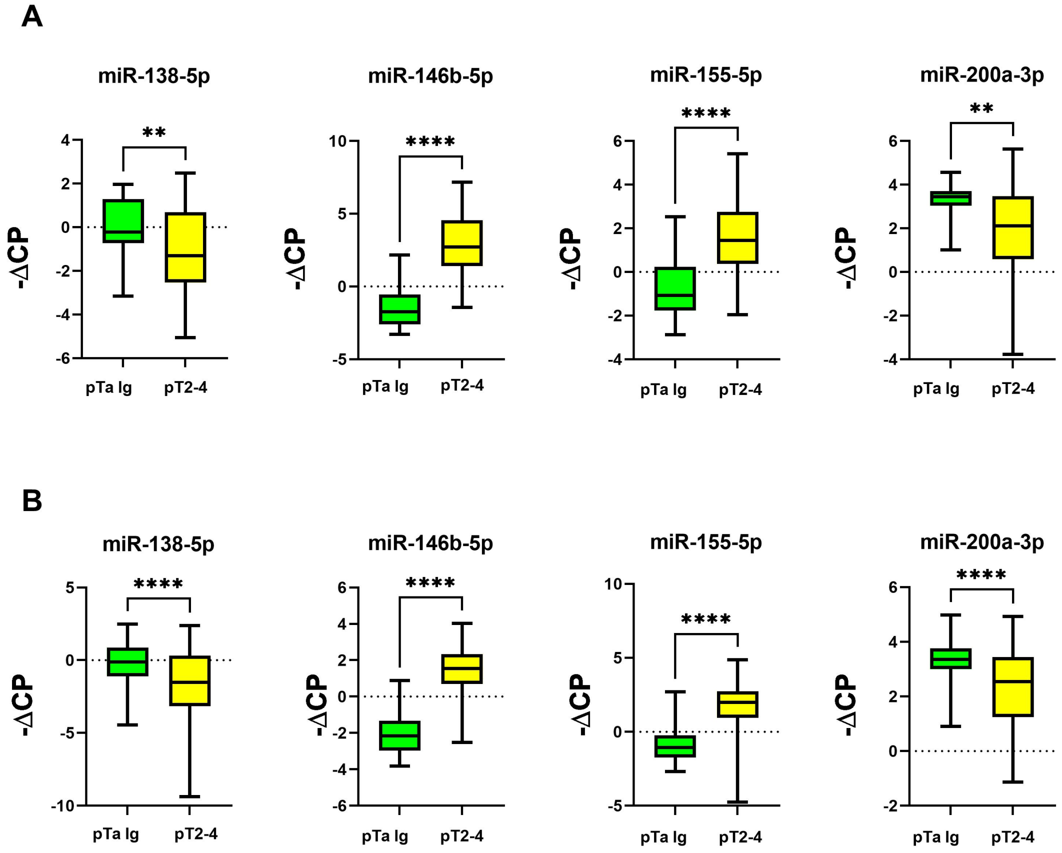


Figure 9.1: Expression of miRNAs in MIBC vs. pTa lg cancers. For each miRNA, its expression is shown as $-\Delta$CP values for the MIBC (denoted as pT2-4) and pTa lg samples in Cohort 1 (A) and Cohort 2 (B). A larger $-\Delta$CP value denotes a higher expression. Significant expression differences between the two groups were identified using Mann-Whitney U tests. The corresponding p-values (p) are annotated as $**$ for p $\leq$ 0.01 and as $****$ for p $\leq$ 0.0001 and can be found in Appendix Table G.3.

To further investigate the ability of individual miRNAs to distinguish MIBC from pTa lg tumors, we computed ROC curves as described in Section 9.1.3. The respective AUC values for each miRNA are shown in Figure 9.2, and the corresponding ROC curves can be found in Appendix Figures G.5 to G.8. For TUR-MIBC samples, miR-146b-5p and miR-155-5p had a high AUC (0.83 to 0.91) in both cohorts, while AUCs for miR-138-5p and miR-200a-3p were notably smaller (0.56 to 0.64). For the CYS-MIBC samples, all AUCs were larger, with the highest AUC of 0.98 for miR-146b-5p in both cohorts.

In summary, these analyses indicate that especially miR-146b-5p and miR-155-5p have strong discriminatory ability within each cohort. In the following section, we train ML models to assess the discriminatory ability of not only individual miRNAs but also miRNA combinations. Additionally, we assess how well models trained on one cohort can make predictions for the other.

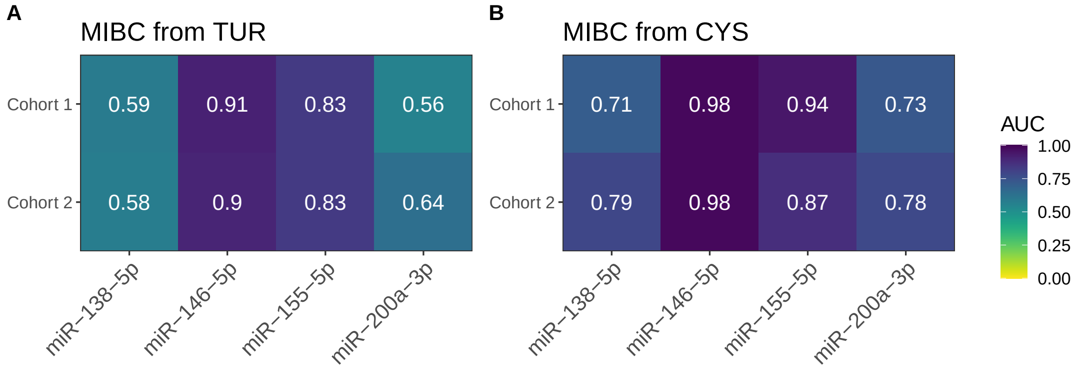


Figure 9.2: AUC values for each miRNA. This figure shows the AUC values for distinguishing MIBC from pTa lg tumors based on each miRNA. Subfigures A and B show the results for MIBC samples obtained via TUR and CYS, respectively.

9.2.2 ML-Based MIBC vs. pTa lg Classification

To determine which combination of the four miRNAs of interest is best suited to distinguish MIBC from pTa lg samples, we trained classification models using five machine learning algorithms: boosting trees, k-nearest neighbors (KNNs), random forests, support vector machines (SVMs), and vanilla neural networks (VNNs). For each algorithm, one model was trained for each possible combination of the four miRNAs as well as each individual miRNA, resulting in a total of $5 \cdot 15 = 75$ models. The black curves in Figure 9.3 depict the performance of these models based on accuracy, sensitivity, and specificity calculated on the test cohort. Here, sensitivity denotes the fraction of correctly classified MIBC samples, and specificity denotes the fraction of correctly classified pTa lg samples. Accuracy is above 0.85 for 40 of the 75 models. For most models with high accuracy, the sensitivity is slightly higher than the specificity. A reason might be the slightly larger number of MIBC ($n = 97$) compared to pTa lg ($n = 86$) samples in the training cohort.

The MIBC samples from the test cohort consist of 16 TUR and 23 CYS probes. Since our major goal is to estimate the risk of muscle invasion before performing a potentially avoidable cystectomy, predictions of our ML models need to be accurate even when only TUR samples are considered. To account for this, we evaluated the ML performance using only the TUR samples of the test cohort. The results are shown as green curves in Figure 9.3. Despite a small decrease in sensitivity, the accuracy across models remains high. Specificity is unaffected since the pTa lg group contains no CYS samples.

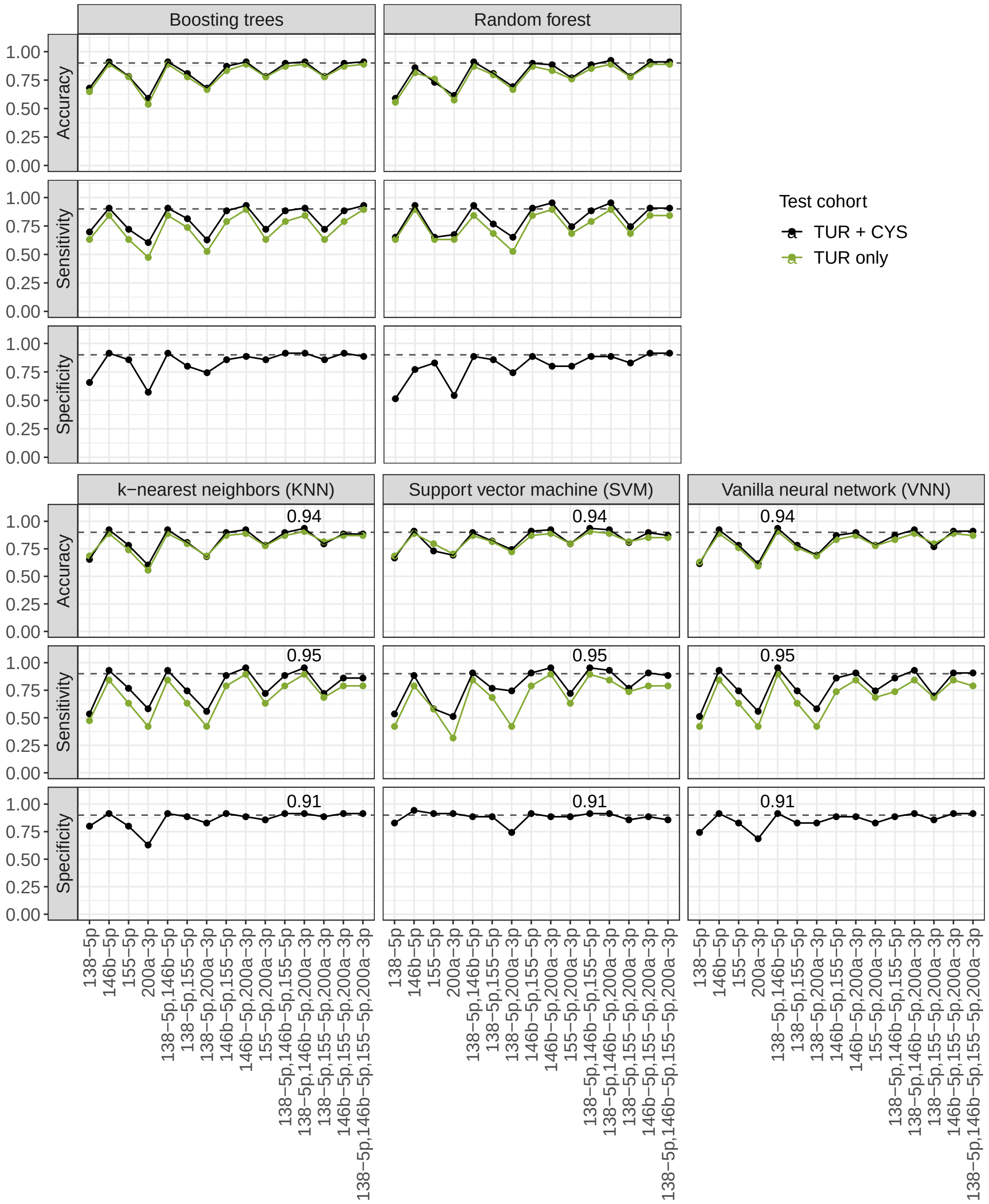


Figure 9.3: Classification performance. This figure depicts the accuracy, sensitivity, and specificity (y-axis) of classification models trained using five ML algorithms and different miRNA combinations (x-axis) as input. While the black curves represent the results for the complete test cohort (TUR and CYS samples), the green curves represent results for TUR samples only. The gray dashed horizontal lines mark an accuracy/sensitivity/specificity of 0.9, respectively. Performance measures for the three best-performing models are annotated numerically.

The overall best classification performance was achieved for three different models:

- a KNN model with miR-138-5p, miR-146b-5p, and miR-200a-3p as features
- an SVM model with miR-138-5p, miR-146b-5p, and miR-155-5p as features
- a VNN model with miR-138-5p and miR-146b-5p as features

All three models have an accuracy of 0.94 (0.91 for only TUR), a sensitivity of 0.95 (0.89 for only TUR), and a specificity of 0.91. In comparison, the AUC varies slightly: 0.97 for the KNN (0.95 for only TUR), 0.96 for the SVM (0.93 for only TUR), and 0.95 for the VNN (0.92 for only TUR). The chosen hyperparameters for each model are listed in Appendix Table G.2.

Figures 9.4A and 9.4B show the expression of miR-138-5p and miR-146b-5p for the training and test cohorts, respectively. Those two miRNAs were included in all of the three best models. In both cohorts, the MIBC and pTa lg groups are well separated with some overlap where the expression value of miR-146b-5p is close to zero. It can be seen that miR-146b-5p alone provides a good separation of the groups, explaining the high accuracy of ML models that only use this miRNA as a predictor (accuracy of up to 0.92). In contrast, miR-138-5p alone does not separate the groups clearly. In Figure 9.4B, the five samples that were misclassified by the best-performing models are highlighted with a thick black border. Interestingly, all three models misclassified the same samples.

Figures 9.4C, D, and E show the predicted class probabilities for the test samples obtained from the best KNN, SVM, and VNN model, respectively. How these probabilities are obtained for the different ML algorithms is detailed in Section 4.2. For the KNN, probabilities of extreme points located at the left and right boundary of the point cloud are 1, and probabilities decrease for samples closer to the center, where a mixing of MIBC and pTa lg samples can be observed. For the VNN, probabilities of 1 are never predicted. Additionally, even points close to the center of the point cloud are often predicted with a high probability above 0.9. For the SVM, the probabilities show slightly more variation than for the VNN but still considerably less than for the KNN. Based on these results, the KNN probabilities mirror the distribution of MIBC and pTa lg as observed in Figure 9.4A and B best.

While class probabilities help estimate model certainty, a model might produce high prediction probabilities but still make many mistakes. Consequently, and especially when ML is supposed to be applied in a clinical context, a guarantee regarding the correctness of predictions is strongly desired. One method to obtain such a guarantee is conformal prediction.

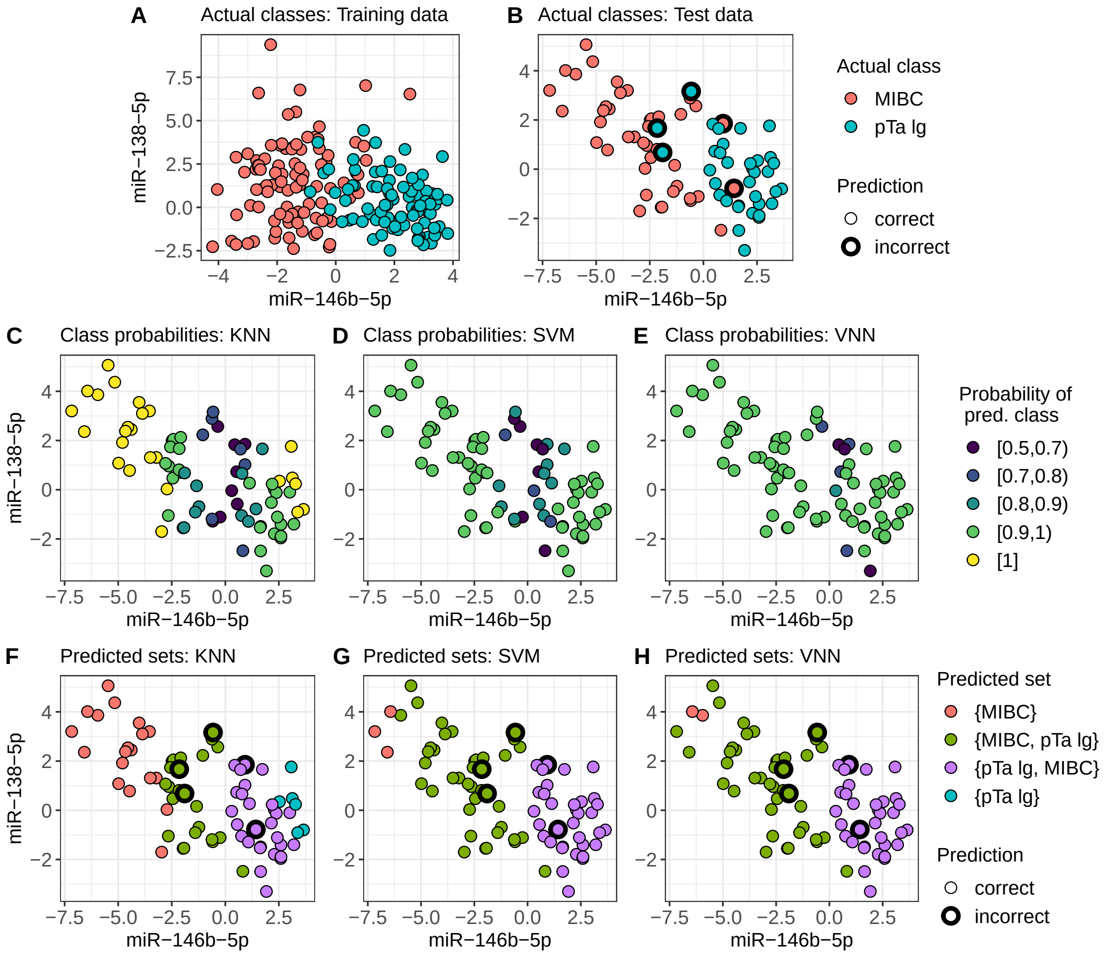


FIGURE 9.4: Predictions of ML models. Subfigures A and B show the expression of miR-146b-5p and miR-138-5p (measured as ΔCP) for the training and test cohorts, respectively. In Subfigure B, samples that were incorrectly predicted by the best-performing KNN, SVM, and VNN models are highlighted with a thick black border. Subfigures C, D, and E depict the predicted class probabilities of the best KNN, SVM, and VNN model, respectively. In Subfigures F, G, and H, the prediction sets obtained by applying conformal prediction are shown.

Figures 9.4F, G, and H show the results of applying conformal prediction using the Summation score described in Section 4.3.2.1 to our data. For samples with high prediction probabilities, the predicted sets only contain a single class and are equal to the predictions obtained without conformal prediction. In contrast, for samples with lower probabilities, conformal prediction yields sets containing both classes, where the more likely class is listed first. These two-class sets mirror the heterogeneity of samples in the area where the expression value of miR-146b-5p is close to 0 and where accurate single-class predictions cannot be guaranteed.

Notably, by predicting two-class sets, conformal prediction also eliminates misclassifications: all five previously misclassified samples are now predicted as two-class sets for the KNN, SVM, and NN. Since no misclassifications remain, the conformal prediction single-class sets contain the correct prediction in every case.

However, the number of single-class predictions is much smaller than the number of two-class predictions for all algorithms, especially for the VNN and SVM. For the KNN, single-class predictions were obtained for 32% of samples. Unlike the KNN, the SVM and VNN never predict the {pTa lg} set and predict the {MIBC} set only for 4% and 3% of samples, respectively. Consequently, the KNN is the preferred model as it yields notably more single-class sets while still eliminating all misclassifications.

In this chapter, we applied conformal prediction using the Summation score. However, in Chapter 7, this score was outperformed by other scores, including the True Class score, which resulted in many more single-class predictions. In Appendix Figure G.9, we show the results of applying the True Class score instead of the Summation score to our bladder cancer models. While the True Class score almost exclusively produces single-class predictions, it fails to eliminate any misclassifications.

9.2.3 Classification of pT1 hg Samples

While our ML models can effectively distinguish MIBC from pTa lg tumors, it would be desirable to accurately predict whether pT1 hg samples are likely to progress to MIBC. As discussed in the introductory section of this chapter, tumors classified as pT1 hg are more aggressive than pTa lg tumors and have already invaded the tissue between the bladder lining and muscle. Consequently, pT1 hg tumors are generally more likely to become muscle-invasive than pTa lg tumors. Accurately predicting disease progression for such cases could support treatment decisions between preventive bladder removal or bladder conservation even when no muscle invasion can (yet) be observed. Unfortunately, follow-up data on disease progression was only available for the twelve pT1 hg samples from Cohort 1 but not for the 108 samples from Cohort 2.

When comparing miRNA expression in pT1 hg tumors from Cohort 2 to pTa lg and CYS-MIBC tumors, miR-146b-5p expression in pT1 hg is significantly different from both. In contrast, the expression of miR-155-5p only differs significantly between pT1 hg and pTa lg, while the expression of the other two miRNAs (miR-138-5p and miR-200a-3p) only differs significantly between pT1 hg and CYS-MIBC. When considering TUR-MIBC samples instead of CYS-MIBC, the same results were obtained for miR-146b-5p and miR-155-5p, but no significant differences were found for the other two miRNAs. The corresponding plots are shown in Appendix Figure G.10.

Out of the 108 pT1 hg samples in Cohort 2, 32% were predicted as MIBC by the best KNN model (see Appendix Figure G.11). However, this result is difficult to interpret without knowing whether the respective samples eventually progressed to MIBC.

For the twelve pT1 hg samples from Cohort 1, follow-up data were available that indicated whether the tumor progressed to MIBC over time. We used the KNN model to make predictions for these samples and compared each prediction to the result of the follow-up: Six of the analyzed tumors progressed to MIBC, five of which were correctly classified as MIBC. However, the KNN also classified five samples as MIBC that did not (yet) show muscle invasion in the follow-up. One of these cases showed a distant tumor metastasis. Unfortunately, due to the limited sample size, we cannot draw generalizable conclusions from these results.

9.2.4 The Role of the Investigated miRNAs in Bladder Cancer

We conducted a literature research to investigate whether our findings for the investigated miRNAs are in line with current biomedical knowledge. All four miRNAs have been extensively studied regarding their role in tumor development and progression in different cancer types, including BC. In Section 9.2.1, we already mentioned that our observations regarding expression differences between MIBC and pTa lg tumors align with the results by Baumgart et al. [429], who investigated the same four miRNAs.

miR-146b-5p: In line with our findings, Xu et al. found a higher expression of miR-146b-5p in advanced stages of BC [441]. The miRNA was also found to promote tumor proliferation and migration [442] but seems to act as both a tumor promoter [443–446] and suppressor [447, 448], depending on the tumor type.

miR-155-5p: Wang et al. found an increased miR-155 expression in higher BC stages and a negative correlation of miR-155 with progression-free survival [449], which is consistent with our results. In vitro experiments using BC cells likewise found that miR-155-5p increases proliferation and invasion [450].

miR-138-5p: Similar to our results, Awadalla et al. found a significant downregulation of this miRNA in MIBC in comparison to non-invasive pT1 tumors [451].

miR-200a-3p: Downregulation of miR-200a-3p was linked to proliferation, migration, and poor prognosis in different tumor types [452–455], which matches our observations. In BC specifically, results have been inconclusive, with some sources reporting downregulation of miR-200a-3p in tumor tissue and urine in advanced tumor stages [446, 447]. Other sources report contradictory results: Wan et al. found miR-200a-3p overexpression to be significantly associated with proliferation, cancer migration, and advanced cancer stages [456]. Similarly, Cavallari et al. found an increased urine level of miR-200a-3p in patients with a high risk of disease recurrence after surgery compared to low-risk patients [457].

9.3 Discussion

A major goal of bladder cancer research is to assess the invasive potential of a tumor to decide if surgical removal of the bladder is necessary or whether the bladder can be conserved. While bladder removal via CYS is the standard for muscle-invasive bladder cancers (MIBC), bladder conservation in combination with TUR is the standard for non-invasive pTa lg and pT1 hg tumors. However, since pT1 hg tumors show the potential to become muscle-invasive eventually, bladder removal might be advised in some cases to prevent tumor spread.

Our analyses confirm the potential of using miRNAs as biomarkers to differentiate pTa lg from MIBC tumors. To prevent bladder removal in cases where muscle-invasion is unlikely, we want to accurately assess the invasive potential of a tumor based on TUR (instead of CYS) samples. Thus, we investigated TUR and CYS samples separately. We found two miRNAs (miR-146b-5p and miR-155-5p) with significant expression differences between pTa lg and TUR-MIBC samples. Thus, we believe detecting muscle invasiveness is possible using solely TUR samples. However, this hypothesis needs to be evaluated in a larger TUR cohort.

Our trained ML models show high accuracy for the pTa lg vs. MIBC classification, even when using only TUR samples for testing. The models also seem robust to data from different sources since they were trained and tested on two different patient cohorts.

Interestingly, miR-155-5p was not contained in two of the three best-performing prediction models despite its strong discriminatory ability in the differential expression and ROC analyses. This observation might be explained by the relatively high expression correlation of 0.62 between miR-155-5p and miR-146b-5p, which is included in all three models. Consequently, both miRNAs might provide similar information but, considered individually, miR-146b-5p seems to be the stronger predictor. In contrast, the other two miRNAs (miR-138-5p and miR-200a-3p) are less predictive on their own but might help classify cases where considering only miR-146b-5p is insufficient.

Additionally, we investigated the predicted class probabilities of the ML models. Especially for the KNN model, class probabilities for misclassified samples were relatively low. This is a desirable model trait since it reflects the model's uncertainty in classifying these samples. Lastly, we applied conformal prediction to our models. Conformal prediction only predicts a single class when the corresponding class probability is sufficiently large. Otherwise, a two-class set is predicted. Consequently, misclassifications can be avoided since the model does not have to decide on a single class in uncertain cases. This is highly desirable for the clinical application of ML models where therapy decision based on erroneous model predictions may negatively impact patient well-being. Indeed, conformal prediction eliminated all misclassifications from the test cohort by predicting previously misclassified samples as two-class sets.

Out of the three best-performing models, the KNN had by far the most single-class predictions. However, even with the KNN model, the majority of samples (68%) was predicted as two-class sets. In contrast to single-class predictions that may reinforce or oppose the opinion of a medical doctor, two-class predictions may indicate cases requiring a careful investigation, e.g., through more complex analytical approaches. Generally, we would like to limit the number of two-class predictions since they provide no clear response on the invasiveness of a sample. However, a decreased number of two-class sets generally comes at the cost of some misclassifications in the single-class predictions. Thus, we may tolerate a larger number of two-class predictions if we can thereby avoid prediction errors. The predicted class probabilities can still be used to infer which class (pTa lg or MIBC) the model deems more likely, even for two-class sets.

To reduce the number of two-class predictions, the conformal prediction error rate α can be increased (cf. Section 4.3.2.1). Here, we used $\alpha = 0.1$ which should guarantee that the predicted sets contain the true class of a sample in $1 - \alpha = 90\%$ of cases. When α is set to a larger value, e.g., 0.3, the number of single-class predictions increases since the model needs to be less certain to predict a single class. However, the certainty guarantee is weaker, since we only guarantee that the true class is contained in the predicted set in 70% of cases. Our analyses also show that the number of single-class predictions can differ strongly between algorithms, even with the same prediction accuracy. Thus, the choice of ML algorithm may improve the number of single-class predictions without the need to increase α. Overall, we are convinced that conformal prediction can enhance trust in ML-based decision support systems by extending prediction models with an indicator for prediction uncertainty.

While our models can accurately distinguish pTa lg and MIBC samples, a major goal would be the prognosis of muscle invasion in pT1 hg tumors. We identified several significant differences in the expression of the investigated miRNAs between pT1 hg tumors and the other two groups. Thus, these miRNAs could be suited to assess the invasive potential for pT1 hg samples. Additionally, our models predicted around one-third of pT1 hg samples as MIBC. However, since the number of data points with available follow-up data was relatively small, we cannot draw strong conclusions on whether the predictions truly reflect tumor progression. Thus, a larger number of pT1 hg samples with follow-up data collected over sufficiently long time periods is required.

The final goal would be to estimate the invasive potential of a tumor not based on TUR or CYS samples but solely based on the patient's miRNA expression in urine. Compared to surgical extraction of a tumor sample, this non-invasive procedure would entail no side effects or recovery times and could be performed regularly. While Baumgart et al. showed that muscle invasiveness could be detected based on urinal miR-146b-5p and miR-155-5p expression [429], this approach needs to be validated in a sufficiently large patient cohort. Ideally, the validation would comprise multiple urine samples at different time points and long-term follow-up analyses. By comparing (1) urine and TUR samples from the same patient, (2) urine samples before and after surgery, or (3) samples of healthy probands compared to BC patients, differences between the various settings could be investigated. We could also investigate whether changes in miRNA expression over time are indicative of a tumor becoming more aggressive or muscle-invasive.

Chapter 10

Summary, Discussion, and Outlook

Nowadays, artificial intelligence (AI) is omnipresent in our everyday lives. AI models achieve impressive results for tasks such as autonomous driving, chatbots, or recommending personalized content on streaming platforms. Undoubtedly, the healthcare sector can also benefit immensely from this technology, e.g., through image-based tumor detection, surgical robots, or AI-based systems for medical decision support. In this thesis, we used AI, specifically machine learning (ML), to develop models that should eventually aid in the personalization of cancer treatment. Our main focus was the task of *drug sensitivity prediction*, where we predict how cancer cell lines respond to treatments with various anti-cancer compounds. In a sensitive field such as healthcare, utmost trustworthiness is required from the used ML systems to avoid harm. Thus, we strived to make our models not only accurate but also interpretable and reliable.

In the following, we first summarize the main contributions of this thesis. Next, we discuss the data we used and the model design choices. We contextualize our approaches with existing work and propose directions for future research. Finally, we discuss what factors beyond trustworthy prediction models need to be considered for the clinical application of AI-based decision support systems, outlining recent advancements and remaining challenges.

10.1 Summary

This thesis addresses various challenges of ML-based treatment optimization in cancer. Chapters 2 to 4 provide a comprehensive background on cancer biology, drug sensitivity testing, and machine learning to contextualize our research, which can be summarized by the seven following contributions.

1. In Chapter 4, we identify three core requirements for trustworthy ML: convincing performance, reliability of individual predictions, and model interpretability. While model performance is evaluated routinely in ML, our extensive review of drug sensitivity prediction literature revealed that prediction reliability has hardly been addressed in this field. Additionally, even though interpretability is commonly recognized as a desirable model trait, the term lacked a clear definition. Thus, we proposed a novel taxonomy that formalizes prevalent meanings of interpretability. The taxonomy should not only refine the definition of interpretability but also highlight strategies for achieving it in future research.

2. As stated above, good performance is a core requirement for a trustworthy ML model. Model performance is heavily impacted by the choice of the ML algorithm and the input features. Since cell line data is typically high-dimensional, performing a dimension reduction (DR) for generating model inputs is usually necessary to counteract the curse of dimensionality. While a plethora of different ML algorithms and DR techniques have been applied for drug sensitivity prediction, a systematic and fair comparison was missing. Thus, Chapter 5 presents the largest drug-sensitivity benchmarking to date, where we systematically compare four ML and nine DR approaches by investigating over 16,000,000 prediction models. Our analyses reveal that relatively simple algorithms like elastic net using relatively few inputs frequently outperform more complex models, even state-of-the-art approaches, in terms of performance and interpretability. Notably, neural networks were not competitive either as prediction algorithm or DR technique, even though they are widely popular for drug sensitivity prediction [25, 33, 34].

3. A known issue with drug screening data is the under-representation of drug-sensitive compared to drug-resistant samples [36–38]. This class imbalance negatively affects the performance of classification models trained on such data [358]. In Chapter 6, we show that this phenomenon also manifests in continuous drug response measures, leading to inaccurate regression predictions for sensitive samples in various ML models. To address this issue, we developed SAURON-RF, a novel prediction model based on random forests that simultaneously performs classification and regression. Compared to conventional ML algorithms, a state-of-the-art random forest approach, and a state-of-the-art deep learning approach (cf. Section 7.2.1.2), SAURON-RF strongly improves discrete and continuous predictions for sensitive samples. Additionally, we show that linking the classification and regression tasks improves predictions for both.

4. While Chapters 5 and 6 were mainly concerned with improving performance, Chapter 7 addresses the demand for reliable predictions: We developed a framework for reliability estimation based on conformal prediction (CP) that, for the first time, provides certainty guarantees for drug sensitivity prediction for classification and regression models. ML models augmented with CP predict sets (classification) or intervals (regression) that are guaranteed to contain the true response of a sample at a user-specified certainty level. Small sets and narrow intervals indicate relatively precise predictions, and set/interval size typically increases for samples where prediction-making is more challenging. We applied our framework to SAURON-RF to demonstrate that CP predictions fulfill the specified certainty guarantee and can even eliminate misclassifications, thereby improving both reliability and performance.

5. A trustworthy ML model should not only perform well, be reliable, and interpretable, but it should also fit the task one wants to solve. For personalized treatment recommendations, the ultimate task is *drug prioritization*, i.e., identifying and ranking effective treatment options for a given sample. Comparing the effectiveness of different treatments requires a sensitivity measure that is comparable across drugs, which is not the case for conventional measures like the IC50 or AUC. Thus, in Chapter 7, we introduce a novel, comparable measure called *CMax viability*. Our measure is designed to estimate the maximum effect that a drug can achieve using clinically feasible concentrations. By combining the CMax viability with SAURON-RF and CP, we are the first to enable reliable drug prioritization.

6. In cancer therapy, it is common to administer not only single drugs (monotherapy) but multiple drugs in combination, which can circumvent treatment resistance and decrease side effects [41, 101]. Thus, Chapter 8 focuses on predicting drug responses for combination therapies. Instead of relying on the widely used synergy scores, our models predict the relative inhibition of cell growth at specific drug concentrations defined in the model input. Our method is the first that can make dose-specific predictions of drug combination sensitivity for previously unseen cell lines and drugs. Furthermore, our approach is highly flexible, as any drug sensitivity or synergy measures can be reconstructed from the model predictions. Additionally, we propose an extension of the CMax viability for combination therapies, which enables the simultaneous prioritization of single- and multi-drug treatments, a major step towards personalized treatment recommendation.

7. Besides drug-based cancer therapy, surgery is frequently used to treat the disease. In Chapter 9, we apply ML and CP to tumor samples from bladder cancer patients

to predict whether a tumor will become muscle-invasive based on miRNA expression. The prediction should aid in the decision between surgical removal of the bladder to prevent invasion or bladder-conserving treatment strategies. We first confirmed that four previously identified miRNAs effectively distinguish muscle-invasive and non-invasive bladder cancers. Using these miRNAs, we trained prediction models that are highly accurate, even when evaluated on an independent patient cohort. Moreover, adding CP to our models eliminated all remaining misclassifications. Since the identified miRNAs are also detectable in urine samples, they may allow predicting disease progression in a non-invasive manner.

10.2 Discussion and Outlook

For each of the contributions listed above, the respective chapter provides a detailed discussion of our approach, findings, and shortcomings. In the following, we discuss and contextualize our research more generally, including remaining challenges and directions for future work. Additionally, we outline several factors besides trustworthy ML models that need to be considered to enable the clinical use of AI-based decision support systems.

10.2.1 Data and Model Inputs

For our analyses presented in Chapters 5 to 7, we used data from the *Genomics of Drug Sensitivity in Cancer* (GDSC) database. The GDSC is one of the largest publicly available cancer cell line panels, comprising around 1,000 cell lines screened against several hundred compounds [191]. In Chapter 8, we additionally employed the DrugComb database, which contains harmonized results of multiple drug-screening cell line datasets [46]. Cancer cell lines are a highly popular model system for studying anti-cancer drug responses. However, as discussed in Chapter 3, any model system can only capture in vivo conditions to a certain degree. For example, cell lines cannot represent the three-dimensional tissue architecture, intra-tumor heterogeneity, and tumor microenvironment. While other model systems, such as patient-derived xenografts or 3D organoid cultures, can overcome some of the limitations of cell lines, their use in ML is mainly hampered by the limited data availability (cf. Chapter 3), which poses a significant obstacle for model training.

To counteract the overall data scarcity, data from different datasets and model systems could be combined using methods like transfer- [458] or meta-learning [459, 460]. This approach may also provide the opportunity to integrate data from model systems with clinical patient data. A challenge might be the harmonization of data derived from different platforms, e.g., drug screening data from different assays or gene expression data from microarrays and RNA-seq [461]. Additionally, when combining data from heterogeneous sources, not all information may be available for all samples. For example, some sources may provide molecular properties of the samples but no drug responses (or vice versa), while other sources may provide only monotherapy responses but no data for drug combinations. To address these challenges, semi-supervised learning may be leveraged [211, 213].

Another challenge may be the disagreement in drug screening replicates from different sources [23, 175, 418]. While some screening methods are known to be more robust than others [418], we cannot be sure which results are truly closest to the actual drug response. To account for the observed variation, one might explicitly incorporate replicate data into the same model, e.g., through using a mixed-effects model [462]. For training and prediction-making, data from higher-quality sources could be weighted more heavily. Additionally, we could explicitly account for the presence of replicates when estimating prediction uncertainty.

Besides accounting for experimental variation, biological variation within a sample should also be considered: a tumor typically consists of various subpopulations of cells with distinct molecular properties and drug responses. Thus, a few approaches for drug sensitivity prediction using single cells have been proposed [463]. They focus largely on transferring knowledge derived from bulk experiments to single cells [463]. Currently, data availability remains a limiting factor for developing and evaluating such single-cell models [463].

The analyses presented in this thesis predominantly focus on gene expression data to characterize samples in the model input. However, the GDSC provides further omics types, including data on mutations, copy number variations, and DNA methylation, which are often employed in other prediction approaches [173, 176, 464]. While gene expression is generally considered the most informative omics type for single-omics drug sensitivity prediction [23, 176, 292, 320, 465, 466], the benefit of additionally considering other omics remains debated: Costello et al. [176], Jang et al. [23], and Chiu et al. [292] report that considering additional omics types may enhance performance in some cases. However, the observed improvements were modest, especially in the latter two studies. This aligns with our findings in Section 5.2.7, where we considered models trained using expression and/or mutation data (cf. also Appendix C.2). Similarly, in my Master's thesis, we found that adding mutation and copy number variation data does not improve over models using solely gene expression data for drug sensitivity prediction [116]. In contrast, Wang et al. found that multi-omics inputs performed notably better than single-omics for deep learning [465]. Ali et al. report that the benefit of using multi-omics data may depend on the used algorithm [467]: They observed only marginal improvements when using a support vector machine but notably larger improvements for a Bayesian model. While Iorio et al. found gene expression to be the most informative data type for pan-cancer analyses, they found genomic features superior in cancer-type-specific models [49]. Their findings also support the idea that different omics may be beneficial for different drugs, a notion shared by Koras et al. [320].

In summary, the choice of algorithm and prediction task seem to impact the benefit of different omics types. Thus, systematic benchmarking should be performed when developing novel models or to investigate whether other inputs may improve existing approaches. While applying our models to additional omics types is straightforward, it significantly increases data dimensionality, and careful dimension reduction is likely required to prevent overfitting.

In addition to different omics, attention should also be given to incorporating known drug response biomarkers from databases such as OncoKB [468] and CIViC [469] into prediction models. An example is our classification approach called MERIDA (not discussed in this thesis), which is based on boolean logic [37]. MERIDA not only considers known markers but can also enforce that they are taken into account for the predictions. Ensuring that predictions align with current pharmacological knowledge is certainly beneficial for model trustworthiness. How this mechanism may be incorporated into other ML models and whether it improves performance remains to be investigated.

In a similar vein to adding response markers, other biological knowledge can be included in a model, e.g., protein interactions or the pathway-membership of genes [177, 294]. Using appropriate methods for model interpretability, this may also enhance trustworthiness. However, when Li et al. investigated several pathway-based prediction models, they found that randomly drawn gene sets performed similarly to actual biological pathways [24]. Thus, they recommend performing ablation studies to verify whether adding biological knowledge truly improves predictions for a given model. Additionally, their findings underline that the mere inclusion of biological knowledge into a model does not imply that explanations derived from it reflect the biological reality. Consequently, explanations obtained from such models – but also ML models in general – should always be carefully validated.

10.2.2 Model Design and Evaluation

There is a plethora of possibilities how cell line data and ML can be used to predict drug responses. However, various design choices affect the applicability and performance of a model in different scenarios. Likewise, various evaluation choices are relevant for assessing a model's performance in a setting that is as realistic as possible.

In Chapters 5 to 7, we trained drug-specific models predicting the response to a single compound. In Chapter 8, we instead trained multi-drug models predicting the response of several drugs which may differ substantially in their mode of action. Single-drug models have the benefit that the inputs can be tailored to the specific drug. In contrast, multi-drug models generally benefit from a larger amount of training data. Furthermore, multi-drug models that employ drug characterizations in their input may use this information to exploit similarities between drugs for learning. Such models can often also be applied to previously unseen drugs, e.g., compounds in development or previously untested drug combinations. While we did not evaluate the drug-blind setting, prediction-making for unseen drugs is known to be challenging [313, 389].

Regarding the performance of single- versus multi-drug models, reports are inconclusive: In Sections 5.2.7 and 7.2.1.2, the investigated multi-drug models were inferior to our single-drug models for both regression and classification. Menden et al. [282] and Haider et al. [470] found that the error between single- and multi-drug models for sensitivity prediction was comparable. In contrast, Wei et al. [471] and Suphavilai et al. [472] found that their multi-drug models outperform single-drug approaches such as elastic net and random forests. However, a fair evaluation between both model types is impeded by the following problem: While drug-specific models are typically evaluated on unseen cell lines, i.e., cell lines not used for model training, this is often not the case for multi-drug models where the input consists of a cell line and drug pair (cf. Table 8.1). Thus, randomly splitting the available cell line-drug pairs into a training and test set does not ensure that cell lines used for testing are not included in the training data. It is well-known that prediction-making for previously unseen cell lines is much more challenging than for cell lines seen during training [313]. Unfortunately, we could not verify whether the multi-drug models in the four studies mentioned above were evaluated in a cell-blind setting.

The cell-blind setting mimics making predictions for a previously unseen patient. We are convinced that evaluating this setting is crucial to model the realistic application of our models for obtaining personalized treatment recommendations. Consequently, all models in this thesis were tested in a cell-blind manner.

In terms of ML algorithms, we investigated several approaches for both classification and regression: elastic nets, boosting trees, random forests, K-nearest neighbors models, support vector machines, and neural networks. Additionally, with SAURON-RF, we developed our own prediction model, a modification and extension of random forests.

A noteworthy finding of our work is that neural networks were outperformed by less complex algorithms in almost all analyses. This is surprising, given that neural networks are extremely popular for drug sensitivity prediction. While neural networks can theoretically model highly complex relationships, they may require large amounts of data to learn the required parameters. Thus, the limited data availability could explain the poor performance of the neural networks in our benchmarking in Chapter 5. However, even though the investigated dataset in Chapter 8 was significantly larger, random forests still outperformed neural networks.

It can be argued that we implemented only relatively simple neural networks. However, especially in Chapter 8, we performed extensive hyperparameter tuning. The smaller number of hyperparameter options investigated in our benchmarking in Chapter 5 is caused by the high runtime of training neural networks, especially since we investigated a large number of settings (179 drugs, nine dimension reduction techniques, and different numbers of input features). Additionally, our findings in Section 5.2.7 suggest that even complex, state-of-the-art deep learning approaches can be outperformed by comparatively simple ML algorithms. Likewise, there have been multiple reports that neural networks do not improve over conventional ML algorithms for tabular data [316–318]. For drug sensitivity prediction specifically, this was confirmed by Menden et al. [282], Park et al. [466] and Li et al. [24]. Partin et al. found that neural networks are only superior in data-rich settings [473]. In contrast, Chang et al. [289], Deng et al. [294], and Chiu et al. [292] report that their neural networks outperform conventional ML methods. Unfortunately, the comparison is complicated by the fact that the investigated neural networks are typically multi-drug models, whereas the other approaches generally make predictions for single drugs. As discussed above, this can skew the evaluation if multi-drug models are not evaluated in a cell-blind setting. Even though Deng et al. specifically describe a cell-blind evaluation strategy, this strategy seems to have not been applied in their comparative analyses.

One could argue that there are application scenarios where cell-blind evaluations are not necessarily required, e.g., when using ML to predict drug responses for known cell lines instead of performing experimental drug screens. However, drug sensitivity prediction is typically advertised as a means to obtain personalized treatment recommendations. Thus, we advocate for more transparency when publishing new approaches and explaining how the chosen evaluation setting suits the advertised application. Overall, to pinpoint which modeling choices actually improve predictions (e.g., chosen ML algorithm, considered omics, single- or multi-drug model), systematic benchmarkings to compare models based on the same datasets and under the same conditions (e.g., cell- or drug-blind) should be conducted.

The increasing data availability may facilitate the development of increasingly complex models. Thus, we need to ensure that their predictions remain interpretable. To this end, mechanisms of post-hoc interpretability, as discussed in Chapter 4, can be employed. While a plethora of explainability techniques have been developed for neural networks [474], there are also several approaches for random forests that we could integrate into our SAURON-RF framework [475]. Besides interpretability, we also identified reliability as a core requirement for trustworthy ML. This thesis focused solely on conformal prediction to achieve reliable predictions. However, reliability can also be addressed through other means, e.g., out-of-distribution estimation [476]. This technique can detect unusual model inputs that differ from the training data. Next, models could be modified to generalize better to out-of-distribution inputs [477], e.g., data from different model systems or even humans.

Independent of the specific design choices, we would like to highlight that the approaches presented in this thesis are purposefully designed to be flexible and re-applicable in different settings. For example, the dimension reduction methods implemented in our benchmarking or the sample weights we implemented in SAURON-RF can be utilized in various ML approaches. Our CP framework and prioritization pipeline can likewise be applied to any ML algorithm that provides a notion of prediction uncertainty. Finally, our approach of predicting relative inhibitions presented in Chapter 8 can be used to derive a plethora of drug sensitivity or synergy measures for both mono- and multi-drug therapies. Thus, we are convinced that our contributions provide a valuable and extendable foundation for developing novel approaches.

10.2.3 Further Considerations for Trustworthy AI in Clinical Applications

While companies and researchers regularly employ ML to develop novel drugs or deepen our understanding of diseases, the use of ML to treat humans in a clinical setting is still limited. Since treatment decisions may significantly impact a patient's well-being, and mistakes may have drastic consequences, ML models applied in this field require strict regulations and careful evaluation.

Criteria for the use of AI in medicine have only been formalized in the recent past: In 2017, the United States Food and Drug Administration (FDA) released their *Digital Health Innovation Action Plan*, including rules for the regulation and approval of digital health technologies such as AI-based systems [478]. Similarly, the European Union (EU) passed the *EU AI Act* in 2024, providing regulations to guarantee the safe, lawful, and ethical application of AI [254].

In this thesis, we focused on developing trustworthy ML models for personalized treatment recommendations that – after further development – could eventually be included in an AI-based decision support system. Following the FDA and EU regulations, a trustworthy prediction model is a key requirement for a clinically applicable treatment recommendation system. However, it is certainly not the only one: In the previous sections, we already mentioned the need to train and evaluate models on adequate datasets. This also includes verifying whether the used data is free of biases, e.g., regarding gender or age. Patient data also underlies strict privacy laws, such that data security must be ensured both during model development and clinical application. Due to the ever-increasing amount of available data, models should be able to incorporate novel data, e.g., through incremental learning [479]. Moreover, mechanisms for continuous model evaluation should be in place.

The most carefully tested system may still be considered untrustworthy if users do not feel confident interacting with it. Consequently, the system needs to be easy to use, and results must be presented clearly and unambiguously. This requires extensive usability studies that consider how information is displayed to users with varying medical backgrounds, such as physicians and their patients.

Even if an AI system is approved for clinical application, it may make mistakes occasionally. Physicians may also over-rely on the suggestions of an AI system and accept incorrect predictions without sufficient scrutiny [480]. While reliability estimation may alleviate this problem to some degree, the question remains of who is at fault for an ineffective or harmful treatment: the treating physician, the company that developed the ML system, or even the patient who consented to ML-based treatment? To answer such questions, the European Commission published a proposal in 2022 detailing how the EU's current liability framework can be extended to account for liabilities caused by the use of AI [481].

Undoubtedly, just like ML systems, human experts also make mistakes. Humans can also not process as large data quantities as AI or reach conclusions as quickly. However, unlike AI models that are often constrained to certain input types and formats, humans can base their decisions on information from manifold, potentially non-digitized sources. Moreover, physicians can naturally engage with patients on a level that machines cannot, e.g., by gracefully communicating an unfavorable diagnosis or by being empathetic when assessing the implications of different treatment options. Thus, medical AI systems and human experts should not be seen as competitors but instead as collaborators with their respective strengths and weaknesses [482, 483].

Overall, AI-based medical decision support has rapidly evolved over the past years. The continuously improving prediction strategies, increasing data foundation, and recently established legal basis could only be achieved through the passionate and interdisciplinary cooperation of medical experts, (bio-)informaticians, biologists, pharmacists, lawyers, ethicists, security specialists, patients, and many others. Observing the current advancements, we are convinced that ML-based treatment personalization is on the brink of becoming an eagerly anticipated reality.

Appendix A

List of Publications

Peer-Reviewed Journal Publications

- Gerstner, N., Kehl, T., Lenhof, K., Müller, A., Mayer, C., Eckhart, L., ... & Lenhof, H.-P. (2020). **GeneTrail 3: advanced high-throughput enrichment analysis.** Nucleic Acids Research, 48(W1), W515-W520.
DOI: 10.1093/nar/gkaa306

- Herbig, N., Düwel, T., Helali, M., Eckhart, L., Schuck, P., Choudhury, S., & Krüger, A. (2020). **Investigating multi-modal measures for cognitive load detection in e-learning.** In Proceedings of the 28th ACM conference on user modeling, adaptation and personalization, 88–97.
DOI: 10.1145/3340631.3394861

- Gerstner, N., Kehl, T., Lenhof, K., Eckhart, L., Schneider, L., Stöckel, D., ... & Lenhof, H.-P. (2021). **GeneTrail: a framework for the analysis of high-throughput profiles.** Frontiers in Molecular Biosciences, 8, 716544.
DOI: 10.3389/fmolb.2021.716544

- Lenhof, K., Gerstner, N., Kehl, T., Eckhart, L., Schneider, L., & Lenhof, H.-P. (2021). **MERIDA: a novel Boolean logic-based integer linear program for personalized cancer therapy.** Bioinformatics, 37(21), 3881-3888.
DOI: 10.1093/bioinformatics/btab546

- Lenhof, K., Eckhart, L., Gerstner, N., Kehl, T., & Lenhof, H.-P. (2022). **Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method.** Scientific Reports, 12(1), 13458.
DOI: 10.1038/s41598-022-17609-x

- Eckhart, L.*, Lenhof, K.*, Rolli, L.-M., Volkamer, A., & Lenhof, H.-P. (2024). **Reliable anti-cancer drug sensitivity prediction and prioritization.** Scientific Reports, 14(1), 12303.
  DOI: 10.1038/s41598-024-62956-6

- Eckhart, L., Lenhof, K., Rolli, L.-M., & Lenhof, H.-P. (2024). **A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.**
  Briefings in Bioinformatics, 25(4), bbae242.
  DOI: 10.1093/bib/bbae242

- Lenhof, K., Eckhart, L.*, Rolli, L.-M.*, & Lenhof, H.-P. (2024). **Trust me if you can: a survey on reliability and interpretability of machine learning approaches for drug sensitivity prediction in cancer.**
  Briefings in Bioinformatics, 25(5), bbae379.
  DOI: 10.1093/bib/bbae379

- Lohse, S., Mink, J., Eckhart, L., Hans, M. C., Jusufi, L., Zwick A., ... & Junker K. (2024). **The impact of the tumor microenvironment on the survival of penile cancer patients.** Scientific Reports, 14(1), 22050.
  DOI: 10.1038/s41598-024-70855-z

- Eckhart, L.*, Rau, S.*, Eckstein, M., Stahl, P. R., Ayoubian, H., Heinzelbecker, J., ... & Junker, K. (2025). **Machine learning accurately predicts muscle-invasion of bladder cancer based on three miRNAs**.
  Journal of Cellular and Molecular Medicine, 29(3), e70361
  DOI: 10.1111/jcmm.70361

Publications in Preparation

- Eckhart, L., Lenhof, K., Herrmann, L., Rolli, L.-M. & Lenhof H.-P. (2025). **How to Predict Effective Drug Combinations - Moving beyond Synergy Scores**.
  Accepted for publication in iScience.
  DOI: 10.1016/j.isci.2025.112622

[*] These authors contributed equally to this work.

Abstracts

- Rau S., Eckhart L., Eckstein M., Stahl P., Heinzelbecker J., Hartmann A., ... & Junker K. (2023). **A 3-miRNA signature defines muscle invasive bladder cancer with high accuracy.** 75. Kongress der deutschen Gesellschaft für Urologie.
- Zohari, F., Eckhart, L., Ayoubian, H., Lenhof, H.-P., Stöckle, M., Heinzelbecker, J., & Junker, K. (2023). **A novel miRNA signature to differentiate metastatic from non-metastatic seminomas.** 29th Meeting of the EAU Section of Urological Research. European Urology Open Science, 56, S84.
  DOI: 10.1016/S2666-1683(23)01168-0
- Eckhart L., Rau S., Eckstein M., Stahl P., Heinzelbecker J., Hartmann A., ... & Junker K. (2024). **A miRNA signature defines invasiveness with high accuracy and is associated with molecular subtypes in bladder cancer.** 39th Annual EAU Congress. European Urology, 85, S1898.
  DOI: 10.1016/S0302-2838(24)01433-7
- Zohari, F., Eckhart, L., Ayoubian, H., Lenhof, H.-P., Bohle, R., Stöckle, M., ... & Junker, K. (2024). **miRNA signature for individual prognosis assessment and therapy decision in patients with Seminoma.**
  76. Kongress der deutschen Gesellschaft für Urologie.
- Zohari, F., Eckhart, L., Ayoubian, H., Lenhof, H.-P., Bohle, R., Bremmer, F., ... & Junker, K. (2024). **Development of a miRNA-based signature for individual prognostic assessment in seminoma patients.** 15. Symposium Urologische Forschung der Deutschen Gesellschaft für Urologie. Die Urologie, Sonderheft 1/2025.
- Heinzelbecker, J., Zohari, F., Eckhart, L., Ayoubyan, H., Lenhof, H.-P., Bohle, R. M., ... & Junker, K. (2025). **Development of an miRNA signature for individual prognostic assessment in seminoma patients.** 2025 ASCO Genitourinary Cancers Symposium, Journal of Clinical Oncology, 43(5), 645-645.
  DOI: 10.1200/JCO.2025.43.5_suppl.645
- Zohari, F., Eckhart, L., Ayoubian, H., Lenhof, H.-P., Bohle, R., Bremmer, F., ... & Junker, K. (2025). **The identification of a novel miRNA signature for the individual diagnosis of seminoma patients.** 40th Annual EAU Congress.

- Zohari, F., Eckhart, L., Ayoubian, H., Lenhof, H.-P., Bohle, R., Bremmer, F., ... & Junker, K. (2025). **A miRNA based signature for the individual assessment of the prognosis of patients with seminoma.** AACR Annual Meeting 2025. Cancer Research, 85(8, Supplement 1), 3340.
  DOI: 10.1158/1538-7445.AM2025-3340

Appendix B

Drug Sensitivity Testing

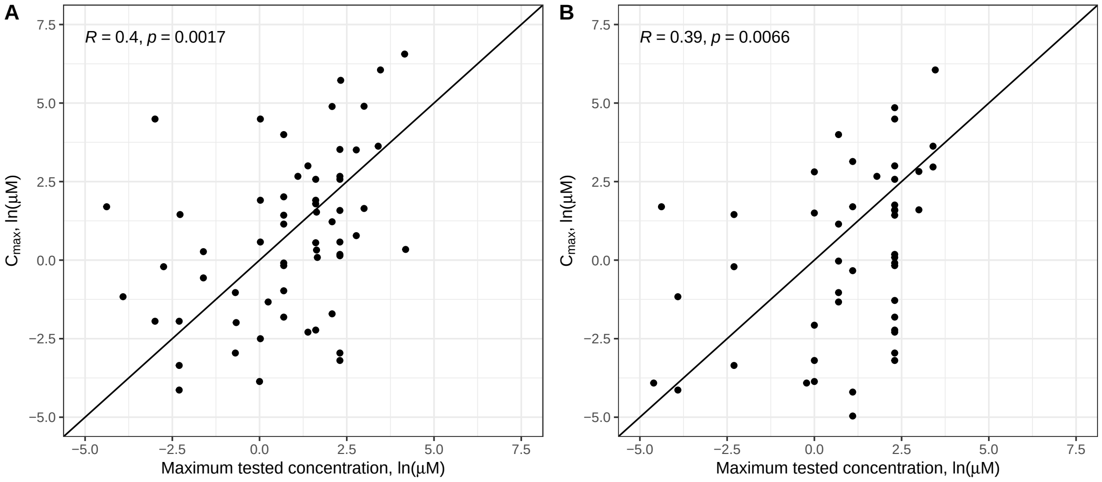


FIGURE B.1: Comparison of CMax values and maximum tested drug concentrations in the GDSC database. This figure depicts the largest screened drug concentrations and CMax concentrations (derived from [174]) for 60 drugs from the GDSC1 database (A) and 47 drugs from the GDSC2 database (B). In the upper left corner of each plot, the Pearson correlation coefficient (R) and corresponding p-value (p) are shown. The black diagonal is the identity function.

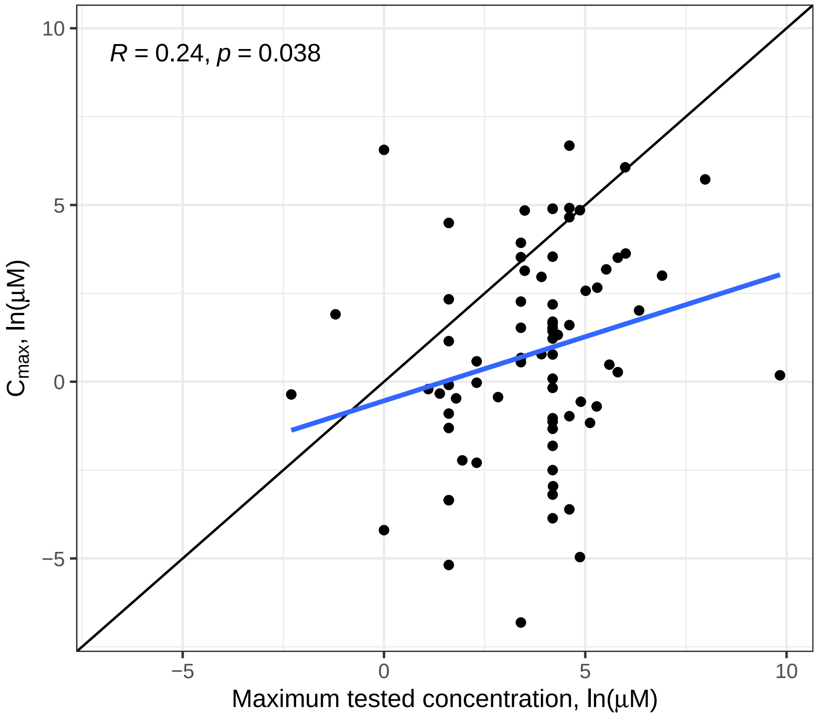


Figure B.2: Comparison of CMax values and maximum tested drug concentrations in the DrugComb database. This figure depicts the CMax concentrations (derived from [174]) for 77 drugs from DrugComb in compared to their maximum screened concentrations. In the upper left corner, the Pearson correlation coefficient (R) and corresponding p-value (p) are shown. Additionally, a regression line (blue) is depicted and the black diagonal denotes the identity function.

Appendix C

Benchmarking

The content and text of this appendix are based on the following publication and its supplement, which were both written by me:

Eckhart, L., Lenhof, K., Rolli, L.-M., & Lenhof, H.-P. (2024). *A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.* Briefings in Bioinformatics, 25(4), bbae242. DOI: 10.1093/bib/bbae242

Table C.1: Overview of all investigated drugs from the GDSC2 dataset with available IC50s for at least 600 cell lines (CLs). Only the 50 drugs with most cell lines were used to train neural networks. Table continues on next pages.

	Drug name	ID	CLs
1	Camptothecin	1003	808
2	5-Fluorouracil	1073	806
3	Afatinib	1032	805
4	Taselisib	1561	805
5	PD0325901	1060	804
6	Linsitinib	1510	804
7	Sapitinib	1549	804
8	Luminespib	1559	804
9	Alpelisib	1560	804
10	SCH772984	1564	804
11	LGK974	1598	804
12	Oxaliplatin	1089	802
13	Irinotecan	1088	801
14	GSK1904529A	1093	801
15	EPZ004777	1237	801
16	EPZ5676	1563	801
17	PLX-4720	1036	797
18	Staurosporine	1034	773
19	Nutlin-3a (-)	1047	773
20	MG-132	1862	773
21	MK-2206	1053	771
22	Trametinib	1372	771
23	Palbociclib	1054	770
24	MK-1775	1179	770
25	Cisplatin	1005	768
26	Docetaxel	1007	766
27	Pictilisib	1058	766
28	AZD7762	1022	764
29	Fulvestrant	1200	764
30	Olaparib	1017	762
31	Dasatinib	1079	760
32	AZD3759	1915	760
33	Vorinostat	1012	758
34	PD173074	1049	758
35	Nilotinib	1013	757
36	Paclitaxel	1080	757
37	Sorafenib	1085	757
38	Dabrafenib	1373	757
39	Lapatinib	1558	757
40	AZD4547	1786	757
41	Gemcitabine	1190	756
42	Bortezomib	1191	756
43	Tamoxifen	1199	756
44	Venetoclax	1909	756
45	Wee1 Inhibitor	1046	755
46	Cytarabine	1006	752
47	Gefitinib	1010	752
48	Dactolisib	1057	752

Continuation of Table C.1.

	Drug name	ID	CLs
49	BMS-536924	1091	752
50	Erlotinib	1168	752
51	YK-4-279	1239	752
52	Epirubicin	1511	752
53	BDP-00009066	1866	752
54	Buparlisib	1873	752
55	Ulixertinib	1908	752
56	AGI-5198	1913	752
57	AZD5363	1916	752
58	AZD6738	1917	752
59	AZD8186	1918	752
60	Osimertinib	1919	752
61	Cediranib	1922	752
62	Ipatasertib	1924	752
63	GDC0810	1925	752
64	GSK2578215A	1927	752
65	I-BRD9	1928	752
66	Telomerase Inhibitor IX	1930	752
67	NVP-ADW742	1932	752
68	P22077	1933	752
69	UMI-77	1939	752
70	Sepantronium bromide	1941	752
71	MIM1	1996	752
72	WEHI-539	1997	752
73	BPD-00008900	1998	752
74	Navitoclax	1011	751
75	Cyclophosphamide	1512	751
76	ABT737	1910	751
77	Afuresertib	1912	751
78	MIRA-1	1931	751
79	Savolitinib	1936	751
80	WIKI4	1940	751
81	Vinblastine	1004	750
82	Temozolomide	1375	750
83	Pevonedistat	1529	750
84	Foretinib	2040	750
85	Pyridostatin	2044	750
86	Vinorelbine	2048	750
87	Ulixertinib	2047	749
88	BIBR-1532	2043	749
89	MK-8776	2046	749
90	Talazoparib	1259	748
91	AMG-319	2045	747
92	VX-11e	2096	746
93	LJI308	2107	746
94	AZ6102	2109	746
95	Rapamycin	1084	745
96	Uprosertib	2106	745
97	GSK591	2110	745
98	AT13148	2170	745
99	VE821	2111	744
100	Dactinomycin	1911	740
101	GNE-317	1926	738
102	Crizotinib	1083	737
103	Uprosertib	1553	735
104	Entinostat	1593	735
105	Alisertib	1051	730
106	Mirin	1048	728
107	Obatoclax Mesylate	1068	728
108	Oxaliplatin	1806	728
109	PRIMA-1MET	1131	728
110	Niraparib	1177	728
111	Fulvestrant	1816	728
112	BMS-345541	1249	728
113	XAV939	1268	728
114	AZD5438	1401	728
115	AZD2014	1441	728
116	AZD1332	1463	728
117	Ruxolitinib	1507	728
118	Leflunomide	1578	728
119	VE-822	1613	728
120	WZ4003	1614	728
121	CZC24832	1615	728
122	PFI3	1620	728
123	PCI-34051	1621	728
124	Wnt-C59	1622	728
125	OTX015	1626	728
126	ML323	1629	728
127	Entospletinib	1630	728
128	PRT062607	1631	728
129	AGI-6780	1634	728
130	Picolinici-acid	1635	728
131	ERK_2440	1713	728
132	ERK_6604	1714	728
133	IRAK4_4710	1716	728
134	JAK1_8709	1718	728
135	VSP34_8731	1734	728
136	Selumetinib	1736	728
137	JAK_8517	1739	728
138	Zoledronate	1802	728
139	Acetalax	1804	728
140	Carmustine	1807	728

Continuation of Table C.1.

	Drug name	ID	CLs
141	Topotecan	1808	728
142	Teniposide	1809	728
143	Mitoxantrone	1810	728
144	Dactinomycin	1811	728
145	Fludarabine	1813	728
146	Podophyllotoxin bromide	1825	728
147	Gallibiscoquinazole	1830	728
148	Elephantin	1835	728
149	Sinularin	1838	728
150	LY2109761	1852	728
151	OF-1	1853	728
152	MN-64	1854	728
153	KRAS Inhibitor-12	1855	728
154	Dinaciclib	1180	727
155	AZD1208	1449	727
156	LCL161	1557	727
157	IWP-2	1576	727
158	I-BET-762	1624	727
159	RVX-208	1625	727
160	GSK343	1627	727
161	AZD5153	1706	727
162	CDK9_5576	1708	727
163	CDK9_5038	1709	727
164	PAK_5339	1730	727
165	TAF1_5496	1732	727
166	IGF1R_3801	1738	727
167	Nelarabine	1814	727
168	ULK1_4989	1733	726
169	Dihydrorotenone	1827	726
170	Sabutoclax	1849	726
171	AZ960	1250	725
172	IAP_5620	1428	725
173	Eg5_9814	1712	724
174	AZD5991	1720	724
175	Ibrutinib	1799	724
176	Vincristine	1818	722
177	GSK2606414	1618	721
178	AZD5582	1617	716
179	Docetaxel	1819	669

Table C.2: Overview of all hyperparameters that were investigated for the training of neural networks.

Parameter	Value(s)
Loss function	MSE
Optimizer	Adam
Learning rate	0.001 (default for Adam)
# Hidden layers	1, 2, 3
# Nodes per layer	input: k, hidden: evenly spaced between in- and output, output: 1
Activation function	tanh, ELU (none in output layer)
Weight initialization	Glorot uniform for tanh activation, He normal for ELU
Bias initialization	0.01
Weight regularization	L2
Bias regularization	none (default)
Dropout	10%, 30%
Batch size	64
Epochs	$\leq$ 4000 (early stopping: 20% of samples as validation data)
Patience	15 epochs

Table C.3: Overview of all hyperparameters that were used for the training of autoencoders.

Parameter	Value(s)
Loss function	MSE
Optimizer	Adam
Learning rate	0.001 (default for Adam)
# Nodes per layer	input: 17,419, hidden (encoder): 3,484 and 697, bottleneck: k, hidden (decoder): 697 and 3,484, output: 17,419
Activation function	ReLU (none in last encoder layer)
Weight initialization	Glorot uniform (default)
Bias initialization	0 (default)
Weight regularization	none (default)
Bias regularization	none (default)
Dropout	none (default)
Batch size	64
Epochs	$\leq$ 100 (early stopping: 20% of samples as validation data)
Patience	5 epochs

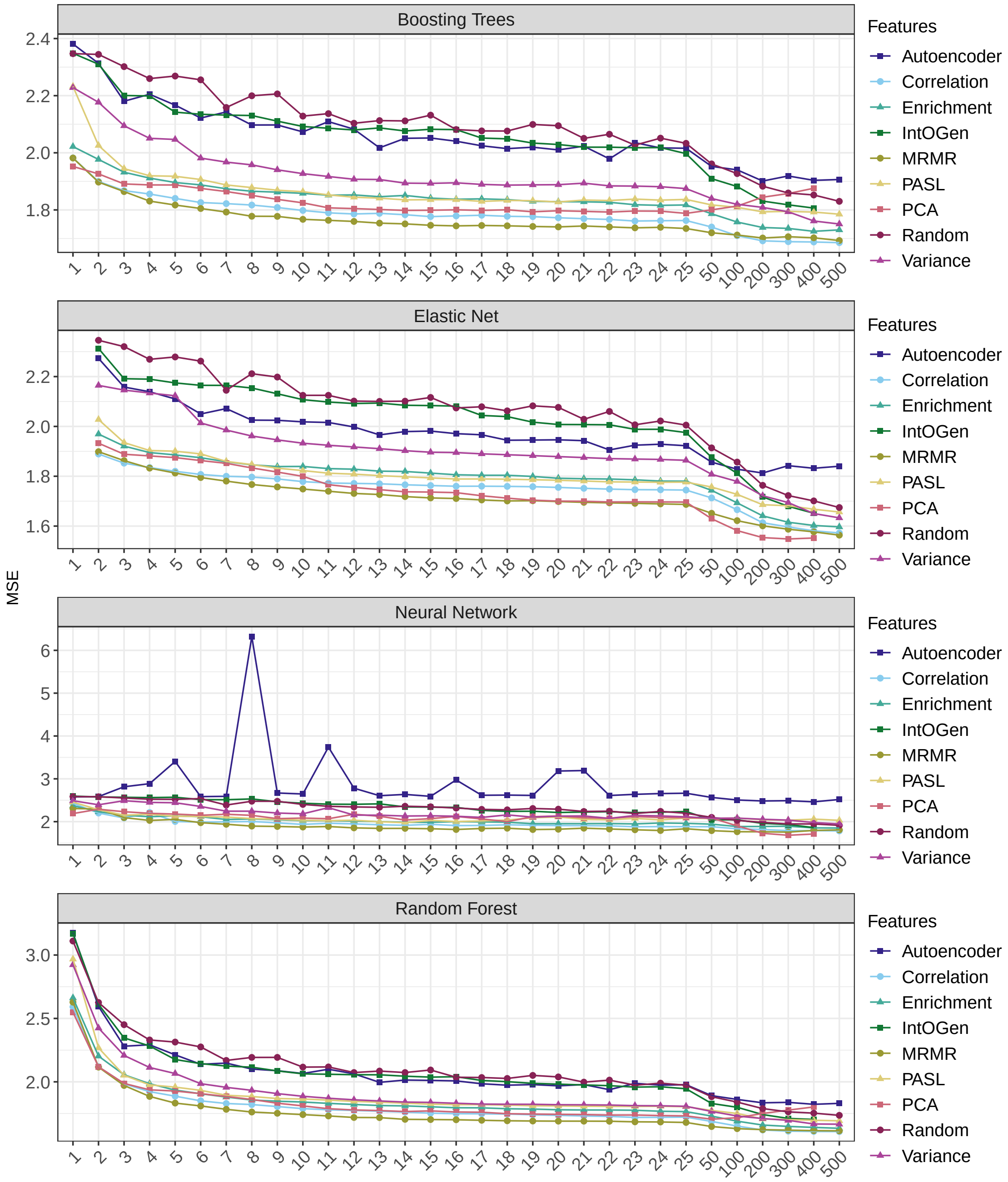


FIGURE C.1: Average test MSEs for each ML method. This figure depicts the test MSEs averaged over all drugs for each combination of DR algorithm, ML method and number of input features. Each plot corresponds to one ML method, where the x-axis denotes the number of input features, the y-axis denotes the mean test MSE, and the coloring represents the different DR techniques. Boosting trees, elastic nets and random forests were applied to all 179 drugs in the GDSC2 dataset for which IC50s for more than 600 cell lines were available. For neural networks, models were only trained on the 50 drugs with most available cell lines (cf. Table C.1).

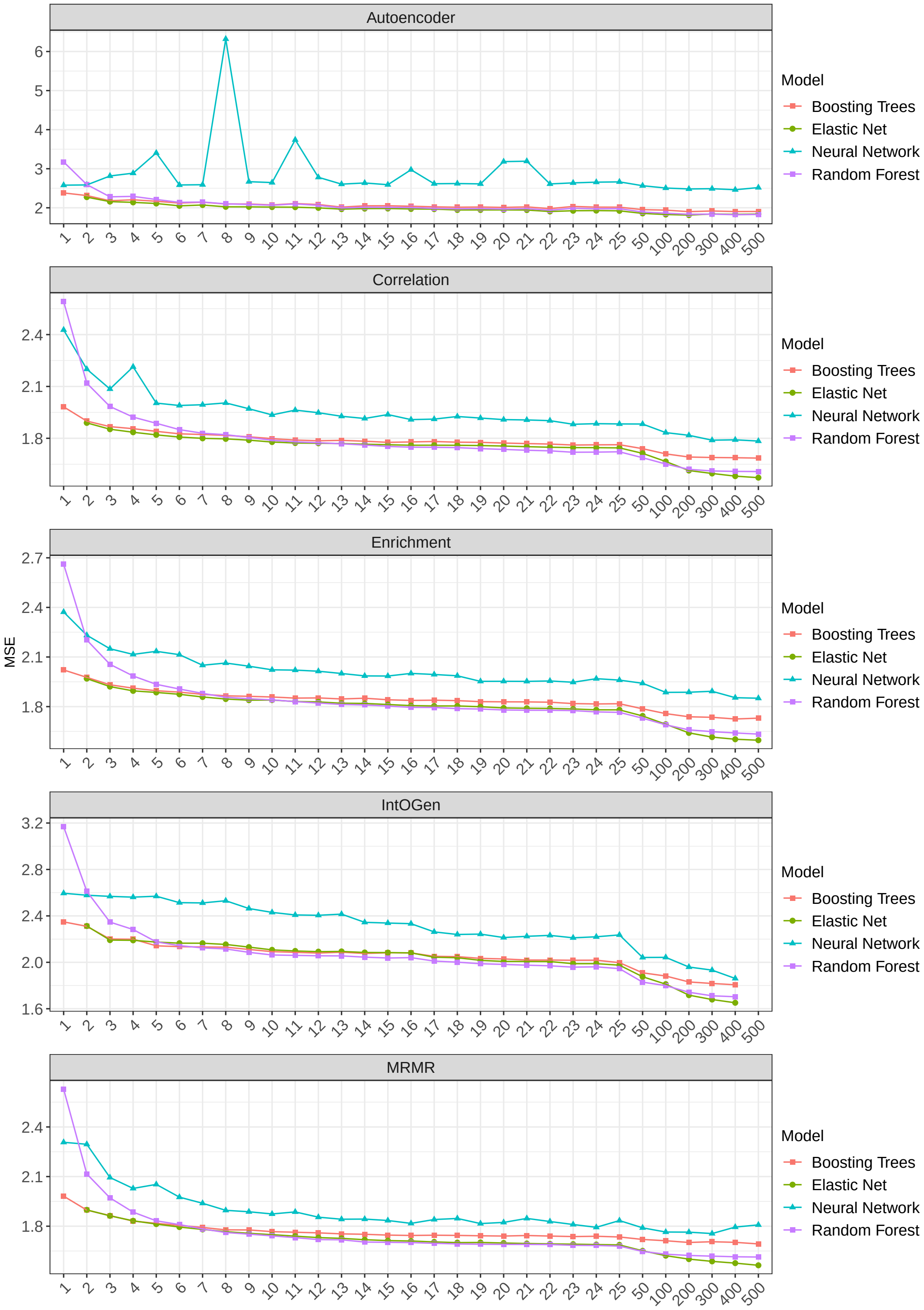


FIGURE C.2: Figure continues on next page.

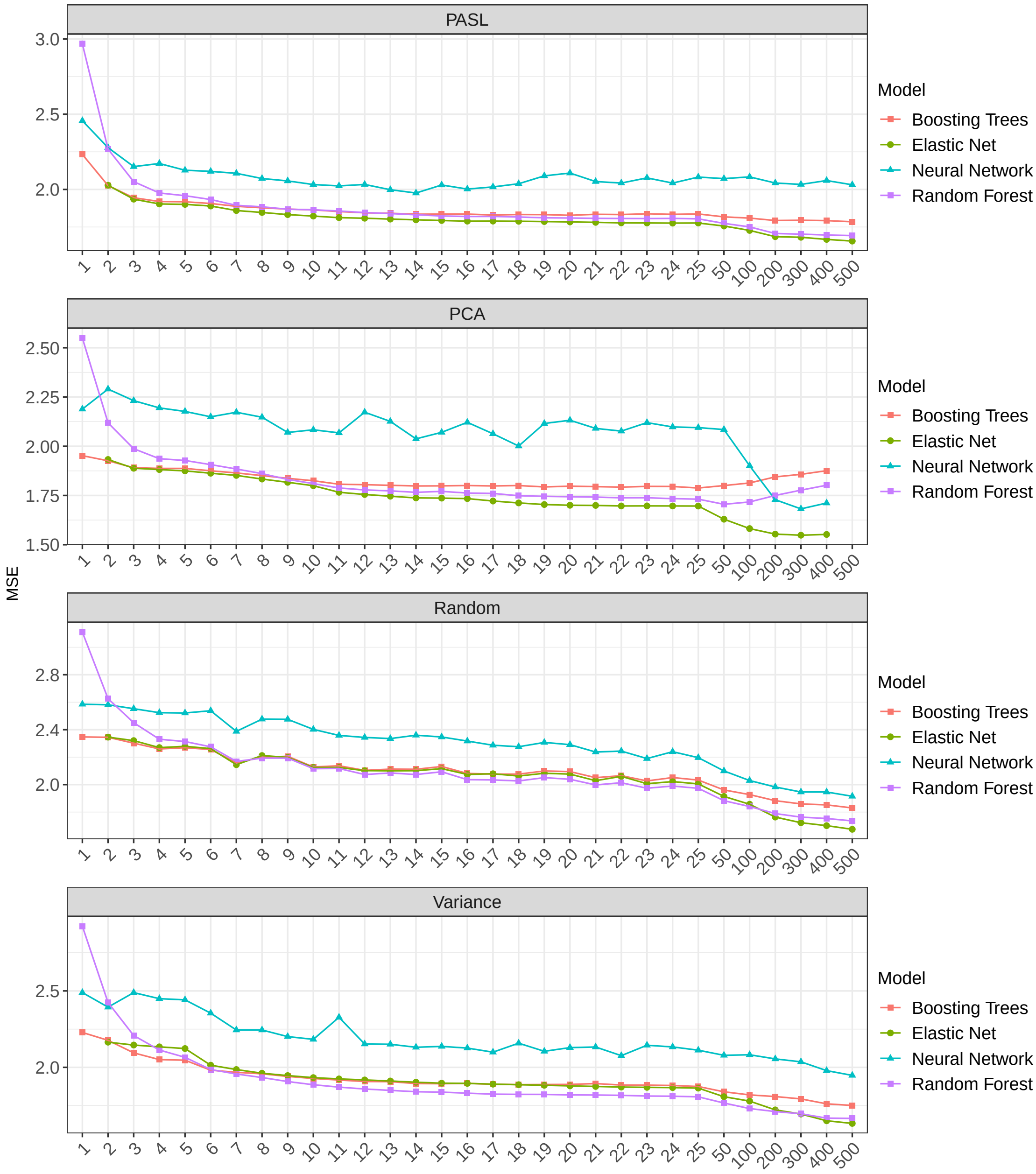


Figure C.2: Average test MSEs for each DR algorithm. This figure depicts the test MSEs averaged over all drugs for each combination of DR algorithm, ML method and number of input features. Each plot corresponds to one DR algorithm, where the x-axis denotes the number of input features, the y-axis denotes the mean test MSE, and the coloring represents the different ML methods. Boosting trees, elastic nets and random forests were applied to all 179 drugs in the GDSC2 dataset for which IC50s for more than 600 cell lines were available. For neural networks, models were only trained on the 50 drugs with most available cell lines (cf. Figure C.1).

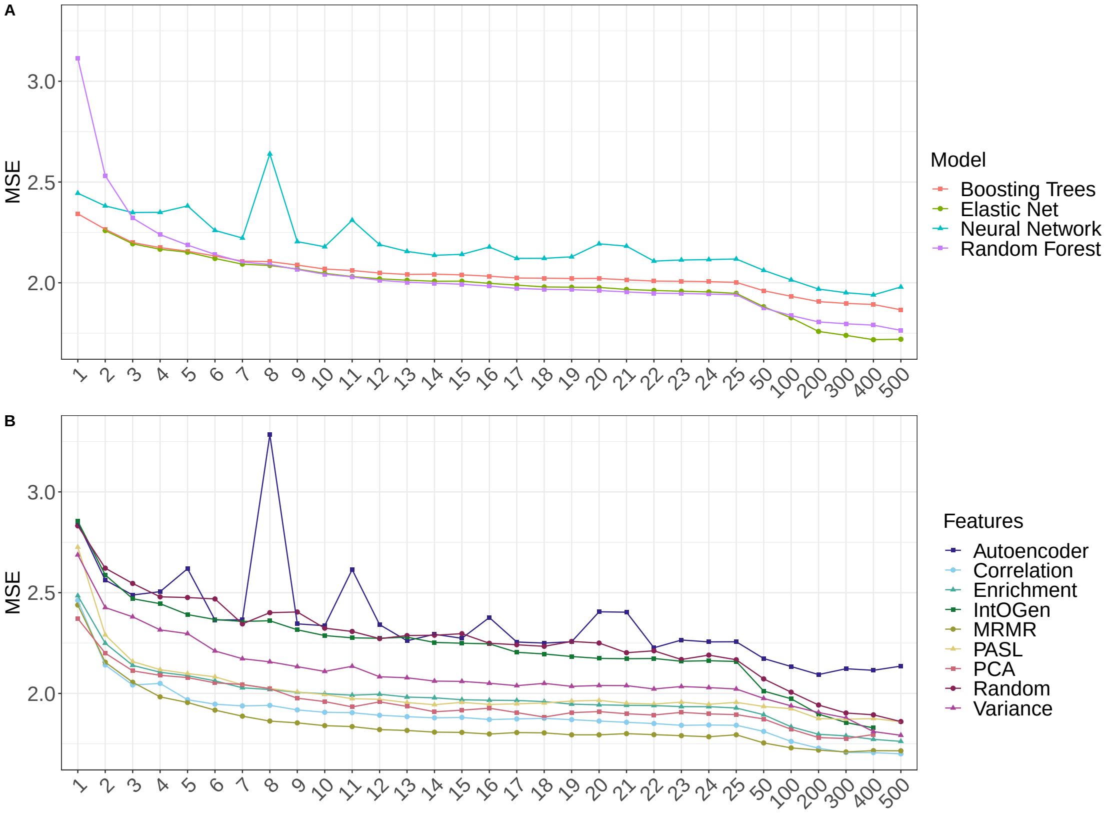


Figure C.3: Average test MSEs for the 50 drugs with most cell lines. This figure depicts the test MSEs averaged over the 50 GDSC2 drugs with most cell lines (cf. Table C.1) for each ML algorithm (A) and DR method (B). The x-axis denotes the number of input features, the y-axis denotes the mean test MSE, and the coloring represents the different ML algorithms or DR techniques.

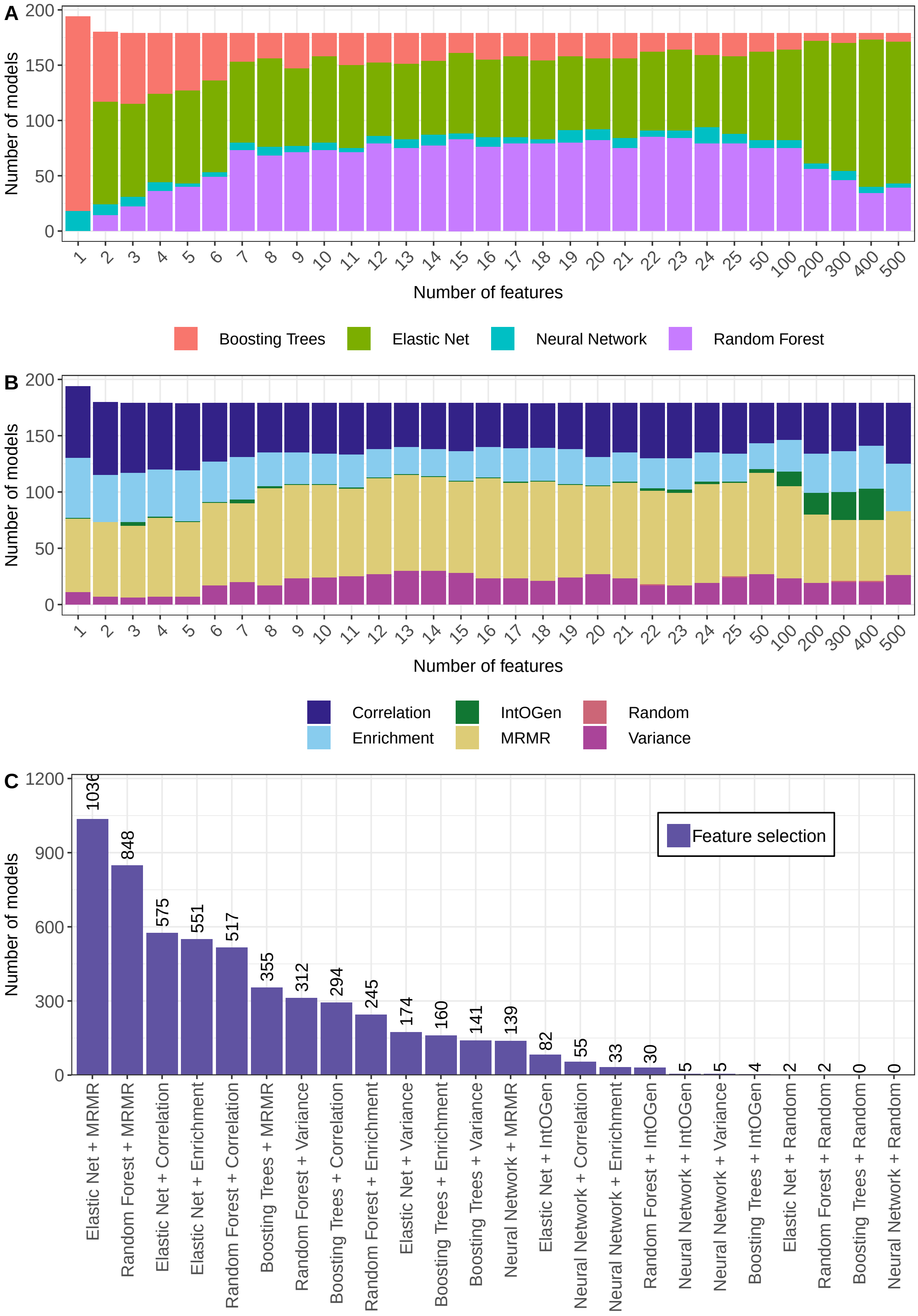


Figure C.4: Best-performing models for each drug and number of input features using only FS methods. Subfigure A and B show how often each ML algorithm and FS method, respectively, yielded the smallest test MSE for each k. Subfigure C shows how often a given combination of ML algorithm and FS method yielded the best results summarized over all k.

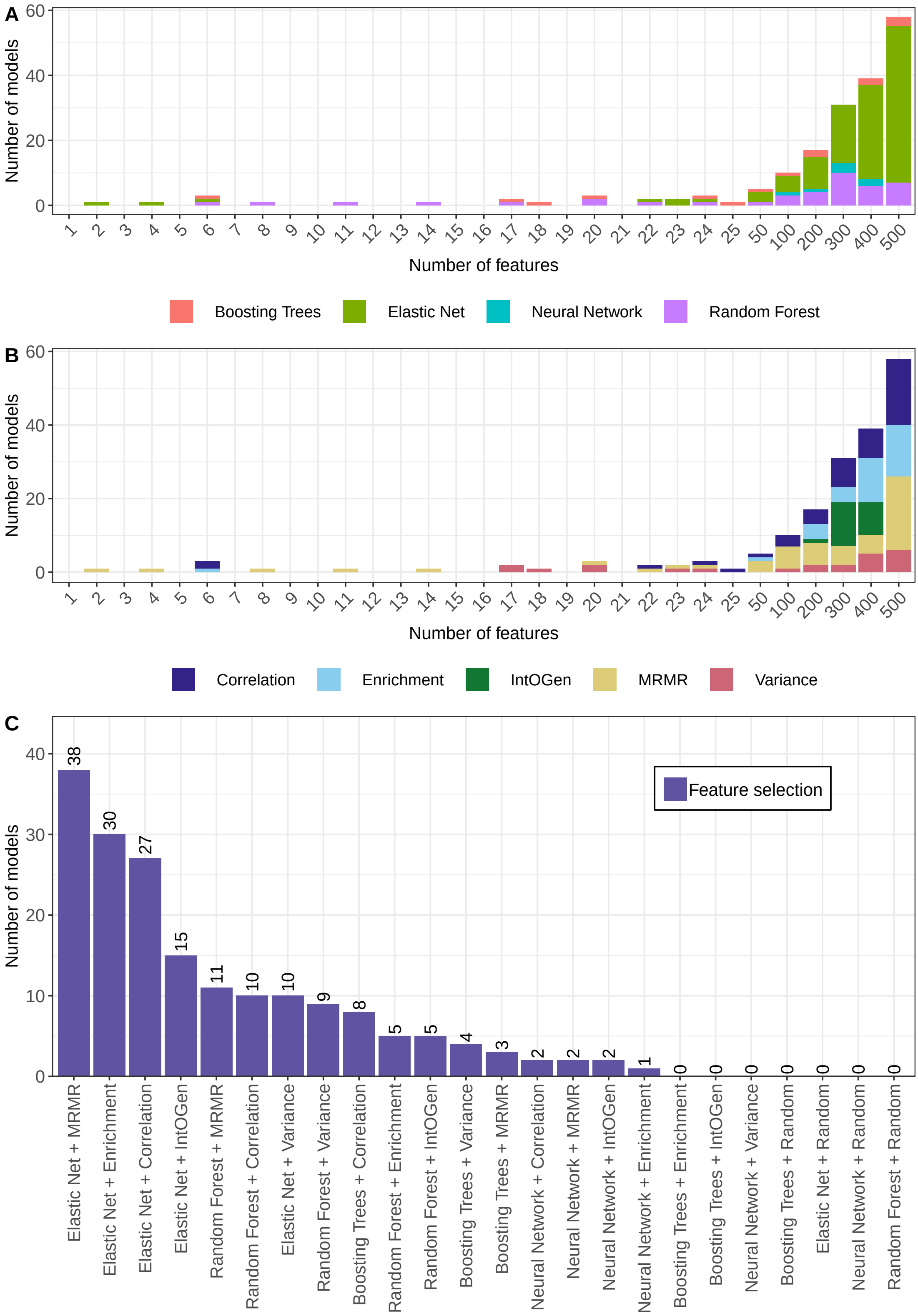


FIGURE C.5: Best-performing models for each drug using only FS methods. Subfigure A and B, respectively, show how often each feature number k and ML algorithm or FS method yielded the smallest test MSE. Subfigure C shows how often a given combination of ML algorithm and FS method yielded the best results.

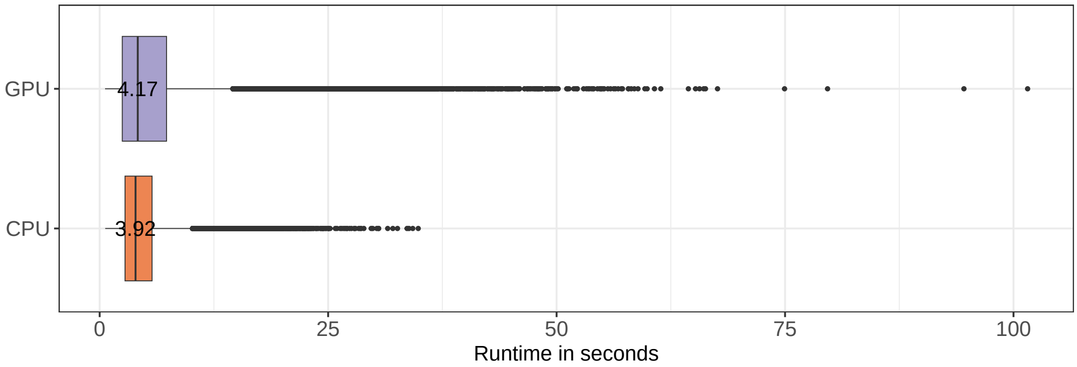


Figure C.6: Runtime comparison GPU vs. CPU. This Figure shows the duration of training neural networks (inputs generated using MRMR FS) using either GPU or CPU. Runtimes are given in seconds and the median runtime is shown in the purple/orange box (mean runtimes are 5.95 seconds for GPU and 4.65 seconds for CPU).

C.1 MRMR Quadratic Program

Let F denote the set of all potential input features and Y the response variable. Furthermore, let k be the number of features to be selected. To measure the dependence between two variables, the mutual information I (cf. Equation 5.6) is used. Gene expression and IC50 values are continuous. Thus, to calculate their mutual information, these measures need to be discretized. To this end, we applied an equal width binning to partition the samples into six bins for each feature and each drug response variable. The quadratic program (QP) can be described as follows: For each feature $X_i \in F$, let x_i be a binary variable that denotes whether X_i is selected ($x_i = 1$) or not ($x_i = 0$). The optimization then selects k features by maximizing the mutual information between the selected features and the response and minimizing the mutual information between selected features:

$$\max_{x} \sum_{i=1}^{|F|} x_i \cdot I(Y; X_i) - \left(\frac{1}{2} \cdot \sum_{j \in \{1,\ldots,|F|\}: j \neq i} x_i \cdot x_j \cdot \frac{I(Y; X_j)}{H(X_j)} \cdot I(X_i; X_j) \right) \tag{C.1}$$

$$\text{such that} \sum_{i=1}^{|F|} x_i = k \tag{C.2}$$

Here, $I()$ denotes the mutual information and $H()$ denotes the entropy.

The ILP was solved using the IBM ILOG CPLEX Optimization Studio V12.6.2 for C++ using 32 cores on an *Intel Xeon Gold 6248* (2.50GHz) CPU. To keep runtime manageable, we limited F to the set of 100 genes for which the mutual information to the investigated drug was largest. Still, we were only able to compute feature sets with $k > 5$ features for a subset of drugs and could not compute any sets for $k > 10$ in a reasonable time (< 500 seconds for a single k on a single training dataset). Figure C.7 shows a performance comparison of the greedy heuristic and the QP.

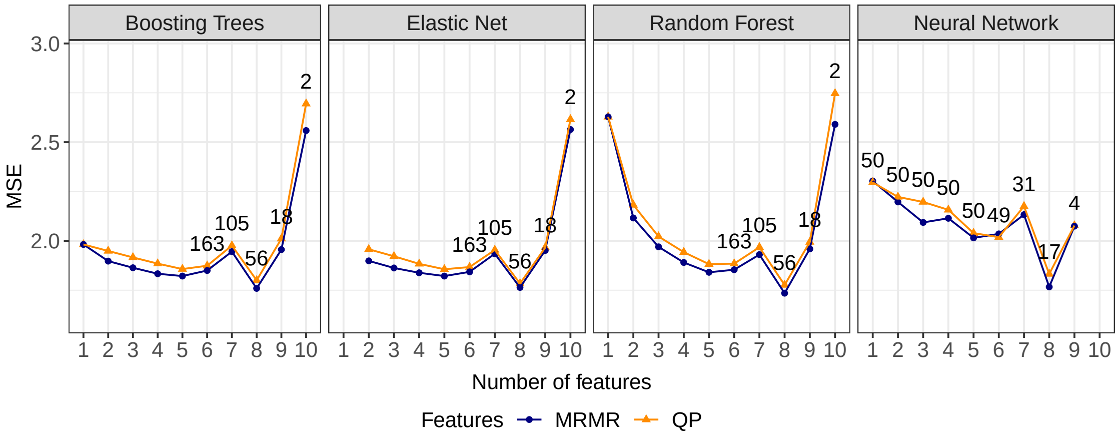


FIGURE C.7: Comparison of QP and heuristic for MRMR feature selection. The average test MSE over all drugs for the best model (i.e., the one with smallest CV MSE) using the respective ML algorithm, feature selection and number of features is shown. As the QP-based features could only be computed for a subset of drugs in the predefined time (< 500 seconds for a single k on a single training dataset), results are only shown for this subset of drugs. Where the number of drugs was smaller than the complete dataset (179 drugs), the data is labeled with the number of drugs over which the average MSE was computed.

C.2 Analyses Using a Multi-Omics, Multi-Drug Deep Learning Approach by Chiu et al.

To investigate the impact of different dimension reduction (DR) procedures on a state-of-the-art method for drug sensitivity prediction and to compare the performance of this method to the ML algorithms discussed in Chapter 5, we performed several analyses using a multi-omics, multi-drug deep learning approach by Chiu et al. [292]. In the following, we will briefly present their approach and then describe the details of our analyses, including the used data, models, and DR techniques. Finally, we discuss the analysis results.

C.2.1 The Approach by Chiu et al.

Chiu et al. developed a multi-omics deep neural network (DNN) for drug sensitivity prediction that predicts the IC50 of multiple drugs simultaneously. The inputs consist of gene expression values and binary mutation data for one cell line. Using one expression-autoencoder and one mutation-autoencoder, these inputs are projected into a lower dimension of $k = 64$ features each. The autoencoders were pre-trained using data from tumor samples obtained from *The Cancer Genome Atlas* (TCGA, `https://www.cancer.gov/tcga`). The pre-trained encoders are then connected to a DNN with drug-specific output nodes. The entire model was trained and evaluated using cell line data from the *Cancer Cell Line Encyclopedia* (CCLE) [484].

C.2.2 Data Processing

To apply the approach by Chiu et al. to the GDSC data and to compare its performance to that of other ML models, we prepared the data as follows:

- *Gene expression data:* We employ the same gene expression data as described in Chapter 5.
- *Mutation data:* We generated a binary mutation matrix $M_{cells \times genes}$, where each entry $M_{c,g}$ denotes whether gene g is mutated in cell line c ($M_{c,g} = 1$) or not ($M_{c,g} = 0$). We obtained coding point mutations of the GDSC cell lines from v99 of the COSMIC cell line project (file: CellLinesProject_GenomeScreensMutant_v99_GRCh37.tsv). In accordance with Chiu et al., we did not consider synonymous mutations.
- *Drug response data:* We employ the same drug response data as described in Chapter 5. However, since the model by Chiu et al. makes predictions for multiple drugs simultaneously, it requires data where each investigated cell line provides IC50 values for each investigated drug. In the GDSC, not all cell lines have been screened against all drugs. To determine a maximal but complete subset of cell lines and drugs for our analyses, we applied an integer linear program (ILP) that can be found in Appendix C.3. This ILP determined a set consisting of 600 cell lines and 170 drugs.
- *Splitting into training and test data:* We randomly split the 600 cell lines with available expression, mutation, and drug response data into a training set (80%) and a test set (20%).

C.2.3 Model Architecture and Hyperparameters

- *Approach by Chiu et al.:* We used the same model architecture and hyperparameters that Chiu et al. employ in their code (`https://github.com/chenlabgccri/

DeepDR). However, we did not only investigate a dimension reduction to $k = 64$ features for each omics type but different feature numbers between 1 and 500 (cf. Figure C.8). Additionally, we investigated the performance when either the expression-encoder or the mutation-encoder was omitted from the model. Note that both the CCLE and TCGA data employed by Chiu et al. measure gene expression using RNA-seq, while the GDSC used for our analyses relies on microarrays. Consequently, pre-training using TCGA data was not possible for our analyses, so we used the training samples for pre-training instead. According to Chiu et al., TCGA pre-trained models resulted in the best performance, but even using randomly initialized encoders outperformed all comparable analyses without pre-training [292]. Consequently, pre-training using the training cell lines should outperform most comparable alternatives including random initialization, when TCGA pre-training is not possible.

- *Elastic net and random forest:* We trained drug-specific elastic net and random forest models using the same training and test cell lines as described in Section C.2.2. We tuned the same hyperparameters as described in Table 5.2 of Chapter 5 using a 5-fold cross-validation on the training data.

C.2.4 Investigated DR Approaches

In addition to the autoencoders employed by Chiu et al., we investigated two further DR methods:

- *Principal component analysis* (PCA): We performed PCA on the gene expression data as described in Chapter 5.
- *Correlation-based feature selection:* For the gene expression features, we employed the same correlation-based feature selection as described in Chapter 5 using Pearson correlation coefficients (PCC). Since the mutation data is not continuous but binary (cf. Section C.2.2), we used Matthew's correlation coefficient (MCC, cf. Equation 4.72) instead of PCC for this data: For each drug, we selected the k genes with the highest absolute MCC between the mutation profile of each gene and the binarized IC50 values of the corresponding cell lines. IC50 values were binarized using drug-specific thresholds obtained from a procedure by Knijnenburg et al. [36] described in Section 3.2.4.2.
- *Multi-drug feature sets:* Since the approach by Chiu et al. is a multi-drug model, we cannot use drug-specific feature sets but need one feature set for all drugs. Since PCA does not make use of any drug response values, it yields the same

features for all investigated drugs, which can directly be used by the approach of Chiu et al. However, we slightly adapted the correlation-based methods presented above: We generated feature sets of different sizes by subsequently including the top $1, 2, ..., f$ most correlated features for each drug, as long as the size of resulting feature sets did not exceed 500. For gene expression data, we were able to include the top $f = 13$ most correlated features for each drug, resulting in 13 feature sets (feature numbers 71, 116, 151, 191, 226, 261, 296, 323, 357, 396, 422, 454, 487). For mutation data, we were only able to include the top $f = 4$ most correlated features for each drug, resulting in four feature sets (feature numbers 136, 253, 364, 454). Since some features are among the top features for multiple drugs, the size of the resulting sets is not necessarily a multiple of the drug number (170).

C.2.5 Results

The results of our analyses are shown in Figures C.8 to C.10. Several observations can be made (see Chapter 5 for further discussion):

- Models based on the approach by Chiu et al. with autoencoders perform worse than all other approaches for most $k \leq 50$. Potentially, these models fail to learn in the number of training epochs chosen by Chiu et al. (100 epochs for autoencoders, 50 epochs for final model) and, consequently, only predict the mean ln(IC50) over all drugs and training cell lines.

- All models based on the approach by Chiu et al. have noticeable discrepancies in the mean test MSE for different k (i.e., the curve shape in Figure C.8 is unstable). A similar phenomenon was also observed for neural networks in our analyses (cf. Figure 5.1 in Chapter 5 and Figures C.1 to C.3 in this Appendix). This might again be caused by network training not converging in the set number of training epochs or by the optimization not being able to leave a local minimum.

- Expression features outperform mutation features for all models (i.e., the models by Chiu et al., elastic nets, and random forests).

- Elastic nets and random forests using mutation features seem to be unable to learn since the test MSE does barely vary across different k.

- Using PCA instead of autoencoders strongly improves the performance of models based on the approach by Chiu et al. for small k. Using correlation-based expression features results in the best test MSEs out of all models based on the approach by Chiu et al.

- Elastic net and random forest models using expression-based features significantly outperform all other models.

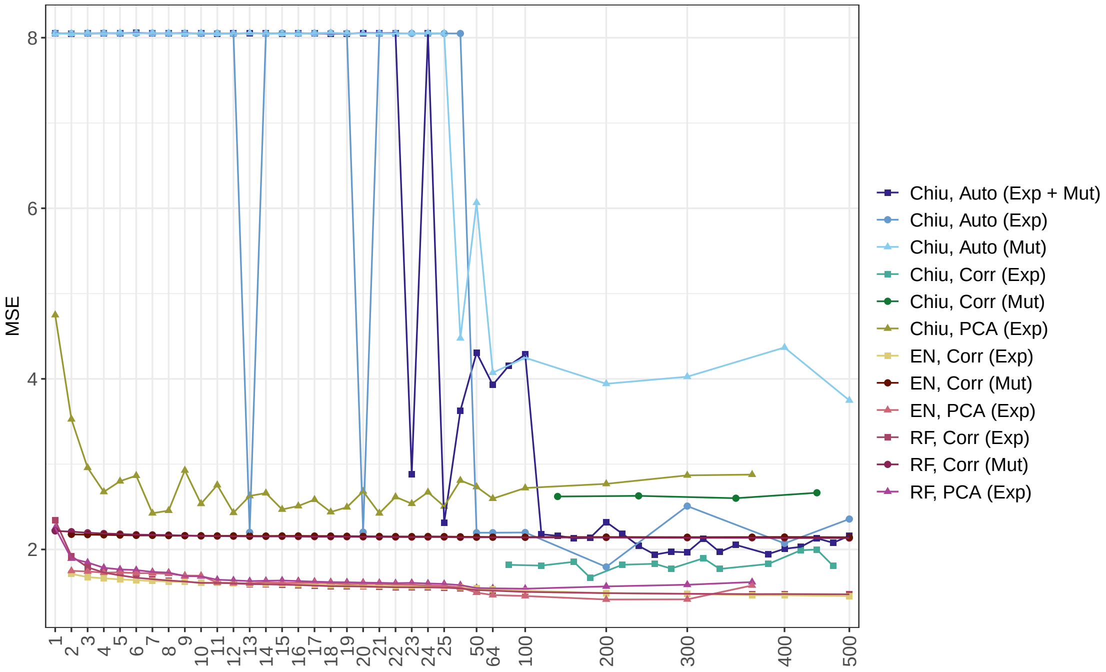


FIGURE C.8: Performance of the approach by Chiu et al. in comparison to other models. This figure depicts the results of applying the prediction approach by Chiu et al. [292] and some variations of it to 170 drugs from the GDSC2. Additionally, the performance of single-drug elastic nets and random forests trained on the same data is shown. The x-axis denotes the number of input features of each data type, the y-axis denotes the mean test MSE averaged over all drugs, and the coloring represents the different approaches. Note that for the multi-omics model by Chiu et al. (Chiu, Auto (Exp + Mut)), the number of features is twice as large as denoted by the x-axis, since two omics types are used. The legend lists all approaches using the following abbreviations: EN - elastic net, RF - random forest, Auto - autoencoder, Corr - correlation, PCA - principal component analysis. In brackets, each model's data types are specified: Exp - gene expression, Mut - mutation.

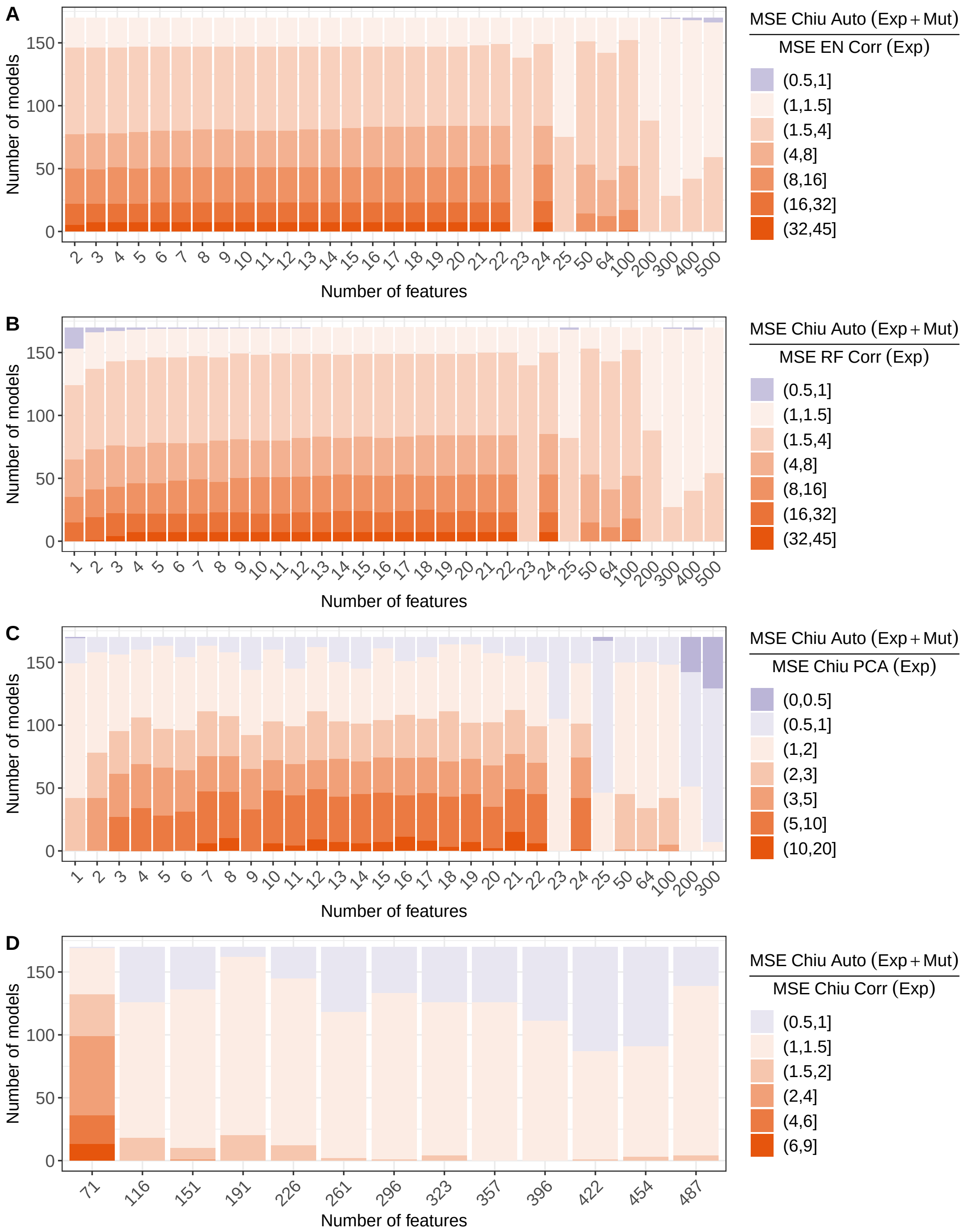


FIGURE C.9: Performance comparison of the approach by Chiu et al. to models with different ML/DR methods. Each subfigure depicts a comparison of test MSEs for two model types, where the MSE for one type is divided by the MSE of the other: Subfigures A and B compare the test MSE of the approach by Chiu et al. for each drug and k to the MSE of drug-specific elastic nets (EN) and random forests (RF), respectively. Both EN and RF were trained using correlation-based gene expression features. Subfigures C and D compare the test MSE of the approach by Chiu et al. to adapted versions of their approach using PCA or correlation-based features instead of autoencoders, respectively. Note that for the multi-omics model by Chiu et al. (Chiu Auto (Exp + Mut)), the number of features is twice as large as denoted by the x-axis, since two omics types are used.

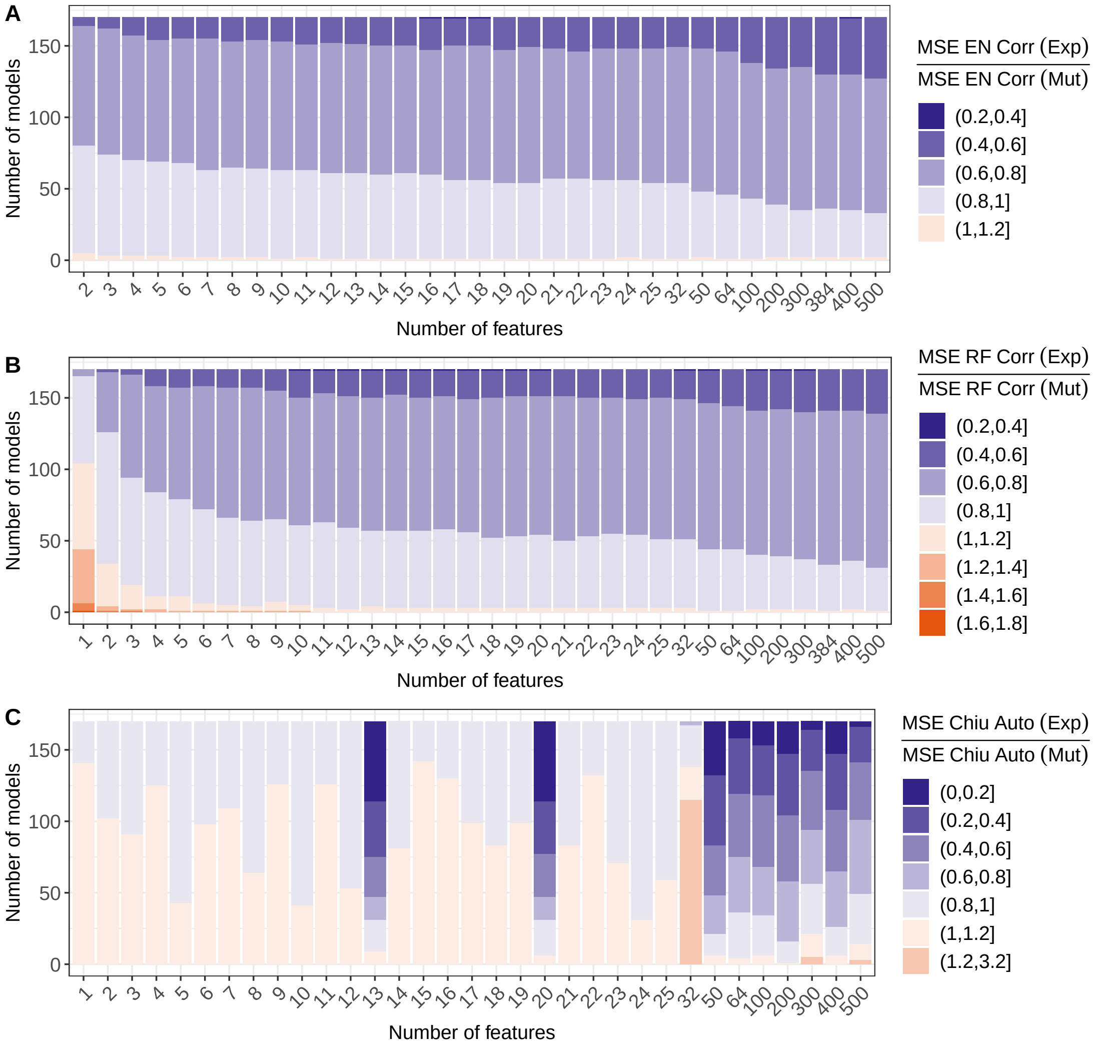


FIGURE C.10: Performance comparison of gene expression and mutation features. Each subfigure depicts a comparison of test MSEs for two model types, where the MSE for one type is divided by the MSE of the other. One model type uses gene expression features, the other uses mutation features. Subfigures A shows results for elastic nets (EN), Subfigure B shows results for random forests (RF), and Subfigure C shows results for the approach by Chiu et al. employing either only the expression encoder or only the mutation encoder.

C.3 Drug and Cell Line Selection Integer Linear Program

We developed an integer linear program (ILP) that determines a maximal subset of cell lines and drugs, where screening data for each cell line and drug combination is available: Let $A \in \{0,1\}^{N \times M}$ denote a binary matrix that indicates for each cell line $i \in \{1, ..., N\}$ and drug $j \in \{1, ..., M\}$ whether drug response data is available for this combination ($A_{ij} = 1$) or not ($A_{ij} = 0$). Next, we define two types of binary decision variables, x_i and y_j, for the chosen cell lines and drugs, respectively:

$$x_i = \begin{cases} 1, & \text{if cell line } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} , \forall i \in \{1, \dots, N\} \tag{C.3}$$

$$y_j = \begin{cases} 1, & \text{if drug } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases} , \forall j \in \{1, \dots, M\} \tag{C.4}$$

The objective function of the ILP is designed to maximize the number of selected drugs and cell lines:

$$\text{maximize} \quad \lambda \cdot (\sum_{j=1}^{M} y_j) + (1 - \lambda) \cdot (\sum_{i=1}^{N} x_i) \tag{C.5}$$

The parameter $\lambda \in [0, 1]$ allows to balance between maximizing cell lines and drugs. Finally, we define several optimization constraints. First, we ensure that cell lines and drugs without available screening data are omitted:

$$\forall i \in \{1, \dots, N\} : x_i \leq \sum_{j=1}^{M} A_{ij} \tag{C.6}$$

$$\forall j \in \{1, \dots, M\} : y_j \leq \sum_{i=1}^{N} A_{ij} \tag{C.7}$$

Next, we ensure that a cell line is not selected if it lacks screening data for any of the selected drugs:

$$\forall i \in \{1, \dots, N\} : \; M \cdot x_i \leq (\sum_{\forall j : A_{ij} = 1} 1) + (\sum_{\forall j : A_{ij} = 0} 1 - y_j) \tag{C.8}$$

In contrast, we ensure that a cell line is selected if it has been screened for all of the selected drugs:

$$\forall i \in \{1, \dots, N\} : \; (\sum_{j=1}^{M} y_j) - (\sum_{j=1}^{M} A_{ij} \cdot y_j) + x_i \geq 1 \tag{C.9}$$

Lastly, we define the minimal number of drugs ($\text{Min}_{\text{drugs}}$) and cell lines ($\text{Min}_{\text{samples}}$) that should be selected:

$$\sum_{j=1}^{M} y_j \geq \text{Min}_{\text{drugs}} \tag{C.10}$$

$$\sum_{i=1}^{N} x_i \geq \text{Min}_{\text{samples}} \tag{C.11}$$

Applying this ILP to our dataset resulted in a subset of 170 drugs with screening data for 600 cell lines using $\lambda = 1$, $\text{Min}_{\text{samples}} = 600$, and $\text{Min}_{\text{drugs}} = 1$.

Appendix D

SAURON-RF

The content of this appendix is based on the following publication and its supplement:

Lenhof, K., Eckhart, L., Gerstner, N., Kehl, T., & Lenhof, H.-P. (2022). **Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method.** Scientific Reports, 12(1), 13458.
DOI: 10.1038/s41598-022-17609-x

Table D.1: Model hyperparameters. Summary of model hyperparameters for different ML algorithms.

Model	Parameter	Value(s)
Boosting Trees	n.trees	1 - 100 in steps of 1
	interaction.depth	4
	shrinkage	0.1
	bag.fraction	0.5
	distribution	“gaussian”
Elastic Net	alpha	0 - 1 in steps of 0.1
	lambda	10^v, v being 100 equally spaced values between -2 and 2
	standardize	TRUE
Neural Network	Loss function	MSE
	Optimizer	Adam
	Learning rate	0.001
	# Hidden layers	1, 2, 3
	# Nodes per hidden layer	same as input layer
	Activation function	tanh (none in output layer)
	Weight initialization	Glorot uniform
	Bias initialization	0.01
	Weight regularization	L2
	Bias regularization	none
	Dropout	10%
	Batch size	128
	Epochs	max. 4000 (early stopping)
	Patience	15 epochs
	Data fraction for validation	20%
Random Forest	n_estimators	500
	min_samples_leaf	15
	max_features (classification)	$\sqrt{\#\text{features}}$
	max_features (regression)	$\frac{\#\text{features}}{3}$
HARF, SAURON-RF	n_estimators	500
	min_samples_leaf	15
	max_features	$\frac{\#\text{features}}{3}$

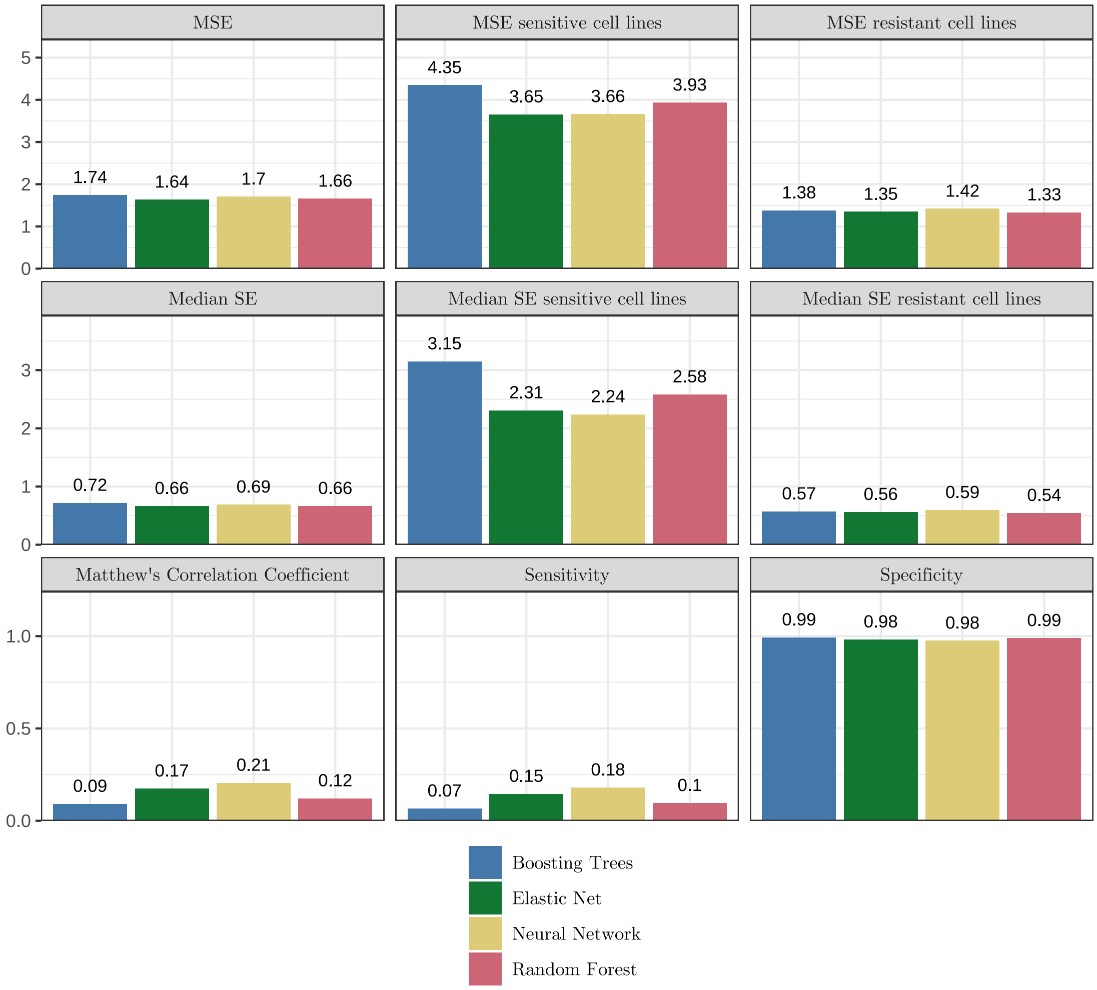


Figure D.1: Test set performance for boosting trees, elastic nets, neural networks, and random forests. The figure shows different regression and classification error measures averaged over 86 drugs.

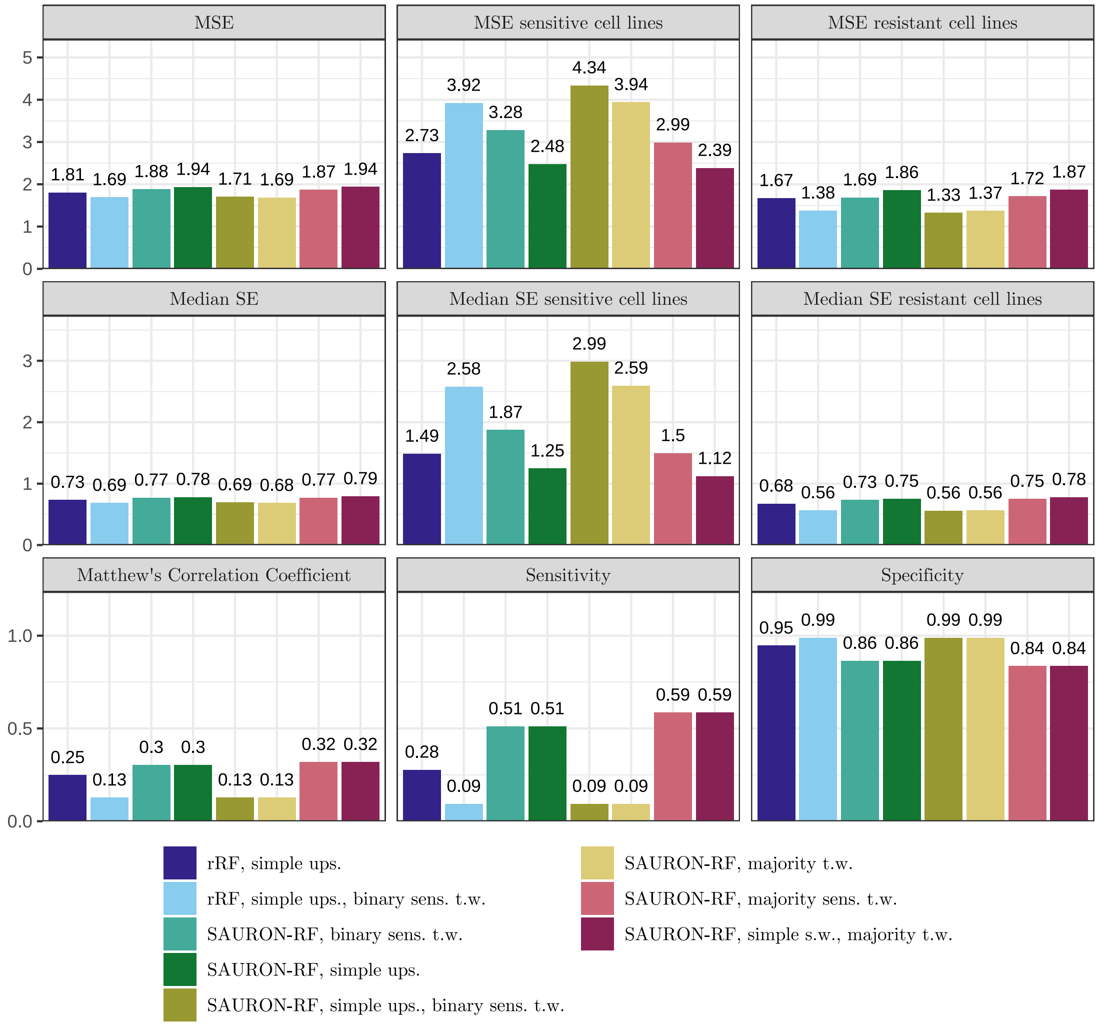


Figure D.2: Test set performance for further versions of regression random forests (rRF) and SAURON-RF. The figure shows different regression and classification error measures averaged over 86 drugs (ups. = upsampling, sens. = sensitive, t.w. = tree weights, s.w. = sample weights).

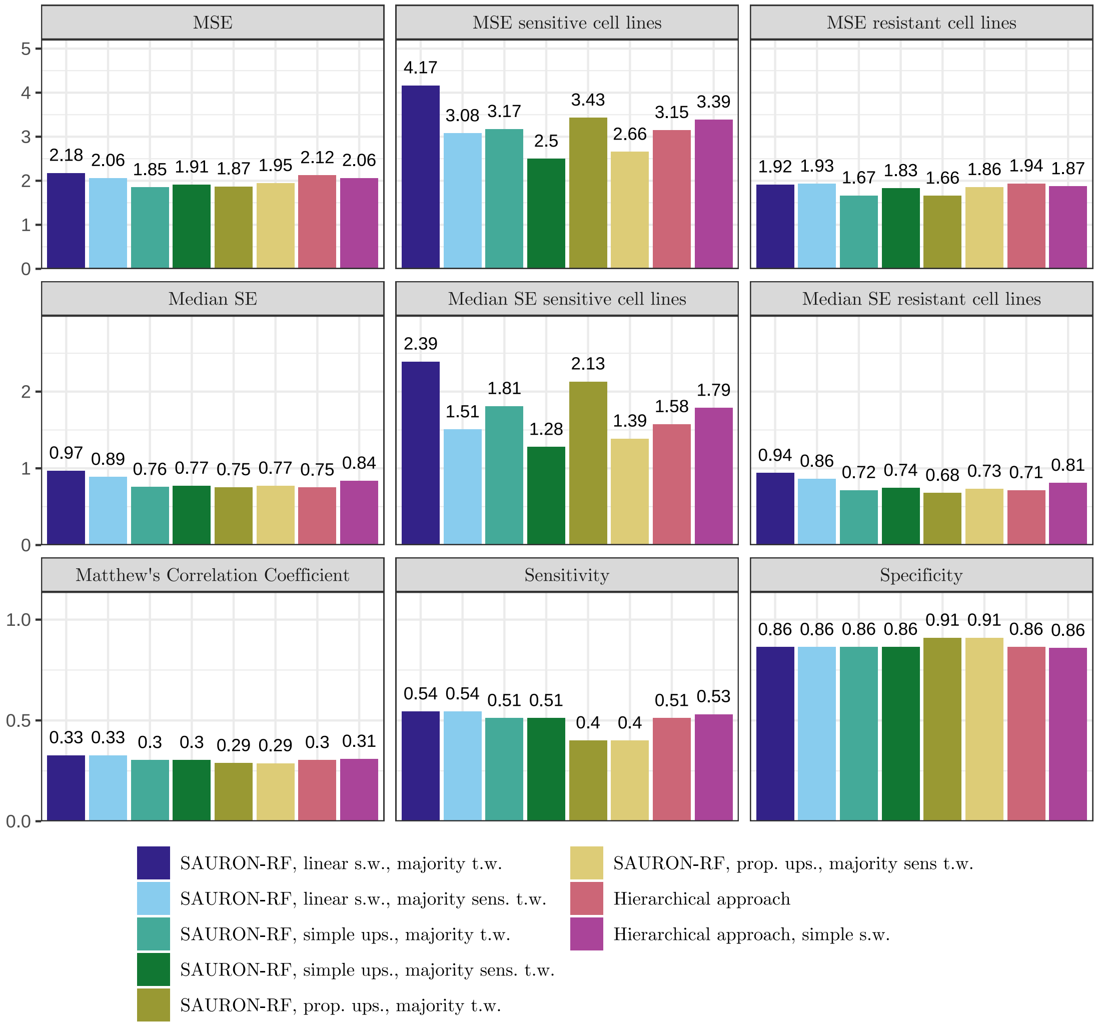


FIGURE D.3: Test set performance for further versions of SAURON-RF and for the hierarchical approach. The figure shows different regression and classification error measures averaged over 86 drugs (s.w. = sample weights, t.w. = tree weights, sens. = sensitive, ups. = upsampling, prop. = proportional).

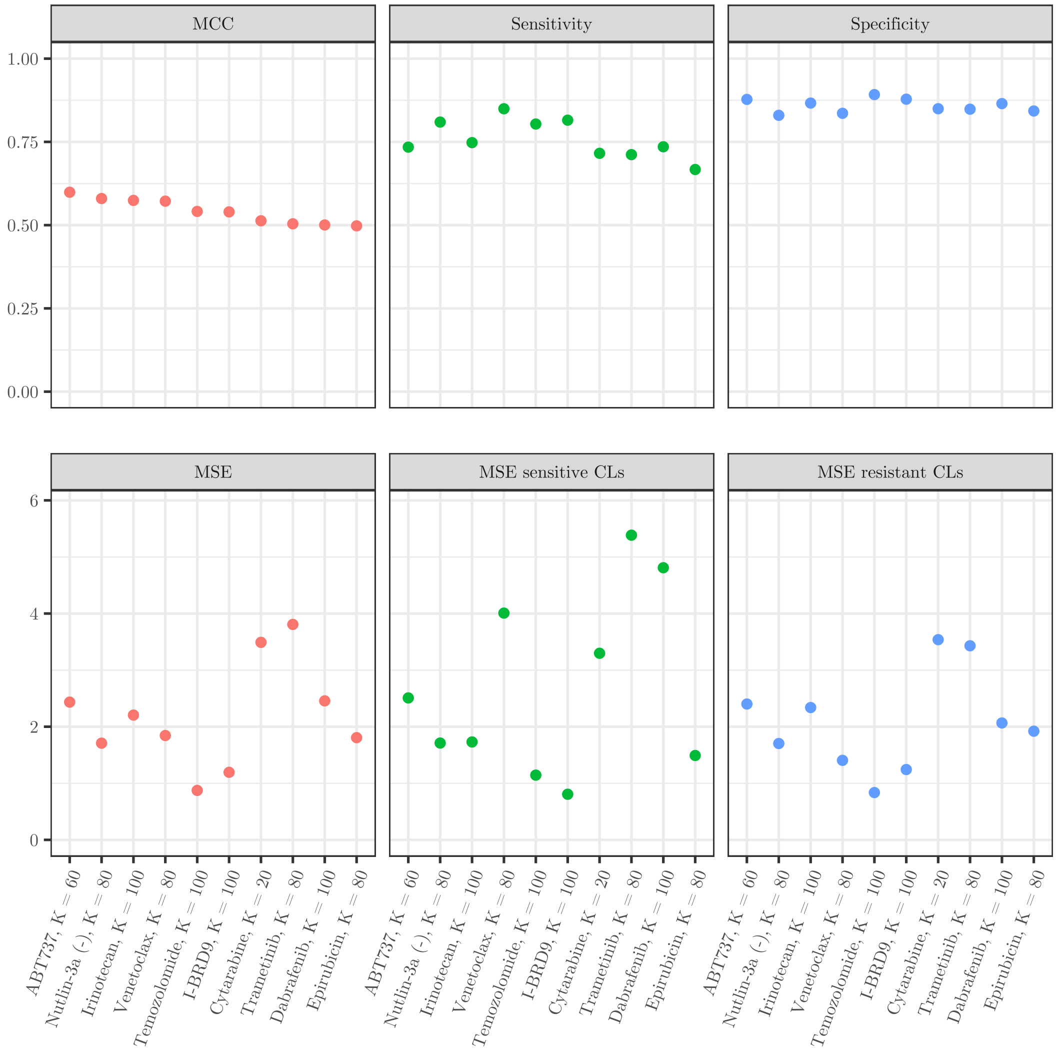


FIGURE D.4: Drugs with best classification performance. This figure shows the classification (top row) and regression (bottom row) performance for those 10 drugs with the highest CV MCC when using the best-performing SAURON-RF model (SAURON-RF with simple sample weights and binary sensitive tree weights) for training. Note that we did not fix the number of input features at $K = 20$, but varied $K \in \{20, 40, 60, 80, 100\}$ and chose that K with the smallest CV MCC.

Drug: ABT737
Target(s): BCL2, BCL-XL, BCL-W, BCL-B, BFL1
Target pathway: Apoptosis regulation

Feature	Feature function (GeneCards [485])	Validated?
TNFRSF12A	involved in extrinsic apoptosis and wound healing regulation	✓ [486]
BCL2	located in the outer mitochondrial membrane and involved in apoptosis	✓ drug target
MIR22HG	involved in wound response	(✓) involved in downregulation of BCL2 ([487, 488])
BLVRB	catalyzes final step of heme metabolism	(✓) low expression associated with obatoclax sensitivity (also BCL2 inhibitor) ([489])
IDH2	catalyzes the oxidative decarboxylation of isocitrate to 2-oxoglutarat	✓ mutations associated with increased sensitivity ([490, 491])

Table D.2: Most important features of ABT737. This table lists the five most important features in the best-performing SAURON-RF model for ABT737 (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

Drug: Dabrafenib
Target(s): BRAF
Target pathway: ERK MAPK signaling

Feature	Feature function (GeneCards [485])	Validated?
BCL2A1	involved in apoptosis	✓ [492]
ARHGAP15	involved in RHO GTPase regulation	×
CTHRC1	may be involved in wound healing	✓ [493]
FAM89A	-	×
MBP	involved in formation and stabilization of myelin membrane	✓ [494]

Table D.3: Most important features of Dabrafenib. This table lists the five most important features in the best-performing SAURON-RF model for Dabrafenib (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

Drug: Nutlin-3a(-)
Target(s): MDM2
Target pathway: p53 pathway

Feature	Feature function (GeneCards [485])	Validated?
MDM2	nuclear-localized E3 ubiquitin ligase, which can promote tumour formation	✓ drug target
RPS27L	might be a component of the 40S ribosomal subunit	✓ [495]
DDB2	part of protein complex that is involved in nucleotide excision repair and cellular response to DNA damage in general	(✓) maybe yes, has to do with MDM2 and nucleotide excision repair [496], Nutlin-3a(-) treatment increases DDB2 expression substantially ([497])
CYFIP2	participates in T-cell adhesion and p53-dependent induction of apoptosis	(✓) Nutlin-3a(-) treatment increases CYFIP2 expression substantially ([498])
SDC4	is a transmembrane proteoglycan involved in intracellular signaling	×

Table D.4: Most important features of Nutlin-3a(-). This table lists the five most important features in the best-performing SAURON-RF model for Nutlin-3a(-) (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

Drug: Trametinib
Target(s): MEK1, MEK2
Target pathway: ERK MAPK signaling

Feature	Feature function (GeneCards [485])	Validated?
DUSP6	has phosphatase activity	✓ [499]
SPRY2	invovled in kinase binding	✓ [500]
BCAS4	part of BLOC-1 complex and associated with breast cancer	×
ETV4	transcription factor	✓ [501]
SLC22A18	has transporter activity	×

Table D.5: Most important features of Trametinib. This table lists the five most important features in the best-performing SAURON-RF model for Trametinib (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

Drug: Irinotecan
Target(s): TOP1
Target pathway: DNA replication

Feature	Feature function (GeneCards [485])	Validated?
SLFN11	involved in tRNA binding, defense response to virus, negative regulation of G1/S transition and replication fork arrest	✓ [502]
SDC4	is a transmembrane proteoglycan involved in intracellular signaling	(✓) high plasma levels of SDC1 related to Irinotecan resistance ([503])
NCKAP1L	transmembrane protein, part of the WAVE complex that regulates cell shape, expressed in haematopoietic cells only	×
DAG1	part of complex that links extracellular matrix to cytoskeleton in skeletal muscle	(✓) combination treatment with PHY906 reduces DAG1 expression compared to other treatments ([504])
CELSR1	it is postulated that the corresponding protein acts as receptor involved in contact-mediated communication	×

Table D.6: Most important features of Irinotecan. This table lists the five most important features in the best-performing SAURON-RF model for Irinotecan (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

Drug: Temozolomide
Target(s): DNA alkylating agent
Target pathway: DNA replication

Feature	Feature function (GeneCards [485])	Validated?
IKZF1	transcription factor	✓ [505]
CYR61	besides others involved in cell proliferation, chemotaxis and angiogenesis	✓ [506]
SDC4	involved in intracellular signaling	×
CTDSPL	has phosphatase activity	✓ host gene of miR-26a, whose overexpression correlates with poor treatment prognosis ([507])
ASPHD2	has metal ion binding activity	×

Table D.7: Most important features of Temozolomide. This table lists the five most important features in the best-performing SAURON-RF model for Temozolomide (cf. Figure D.4). Additionally, the drug target(s) and target pathway as provided by the GDSC database are shown.

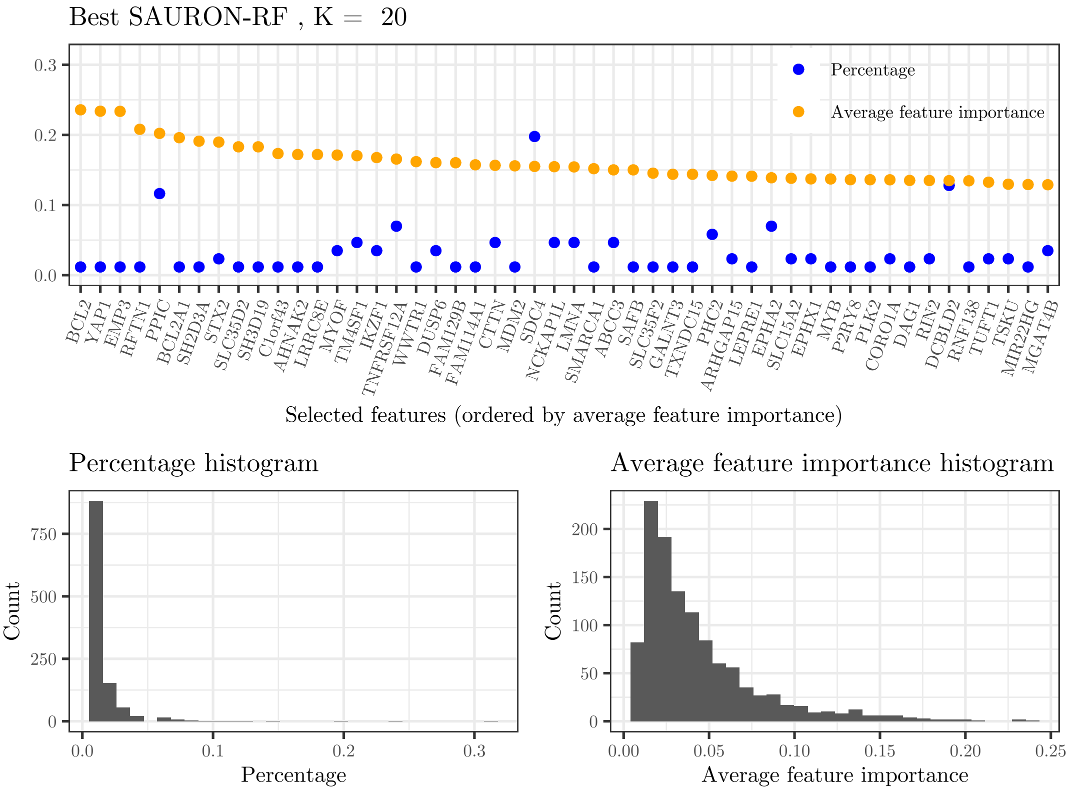


FIGURE D.5: SAURON-RF feature importance. The upper row of this figure shows the 50 genes with the highest mean feature importance for the best-performing SAURON-RF model. Importances are averaged across all drugs for which the respective feature was selected. The blue dots denote the percentage of drugs for which the feature was selected, the yellow dots denote the average feature importance. The lower row of this figure shows feature distributions for the percentage of drugs and the average feature importance, respectively.

Appendix E

Reliable SAURON-RF

The content of this appendix is based on the following publication and its supplements:

Eckhart, L.*, Lenhof, K.*, Rolli, L.-M., Volkamer, A., & Lenhof, H.-P. (2024). *Reliable anti-cancer drug sensitivity prediction and prioritization.* Scientific Reports, 14(1), 12303. DOI: 10.1038/s41598-024-62956-6
(* authors contributed equally)

The descriptions of the approach by Chiu et al. [292] in Section E.4 are based on the following publication and its supplement (where we analyzed the same approach, cf. Section C.2), which was written by me:

Eckhart, L., Lenhof, K., Rolli, L.-M., & Lenhof, H.-P. (2024). *A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction.* Briefings in Bioinformatics, 25(4), bbae242. DOI: 10.1093/bib/bbae242

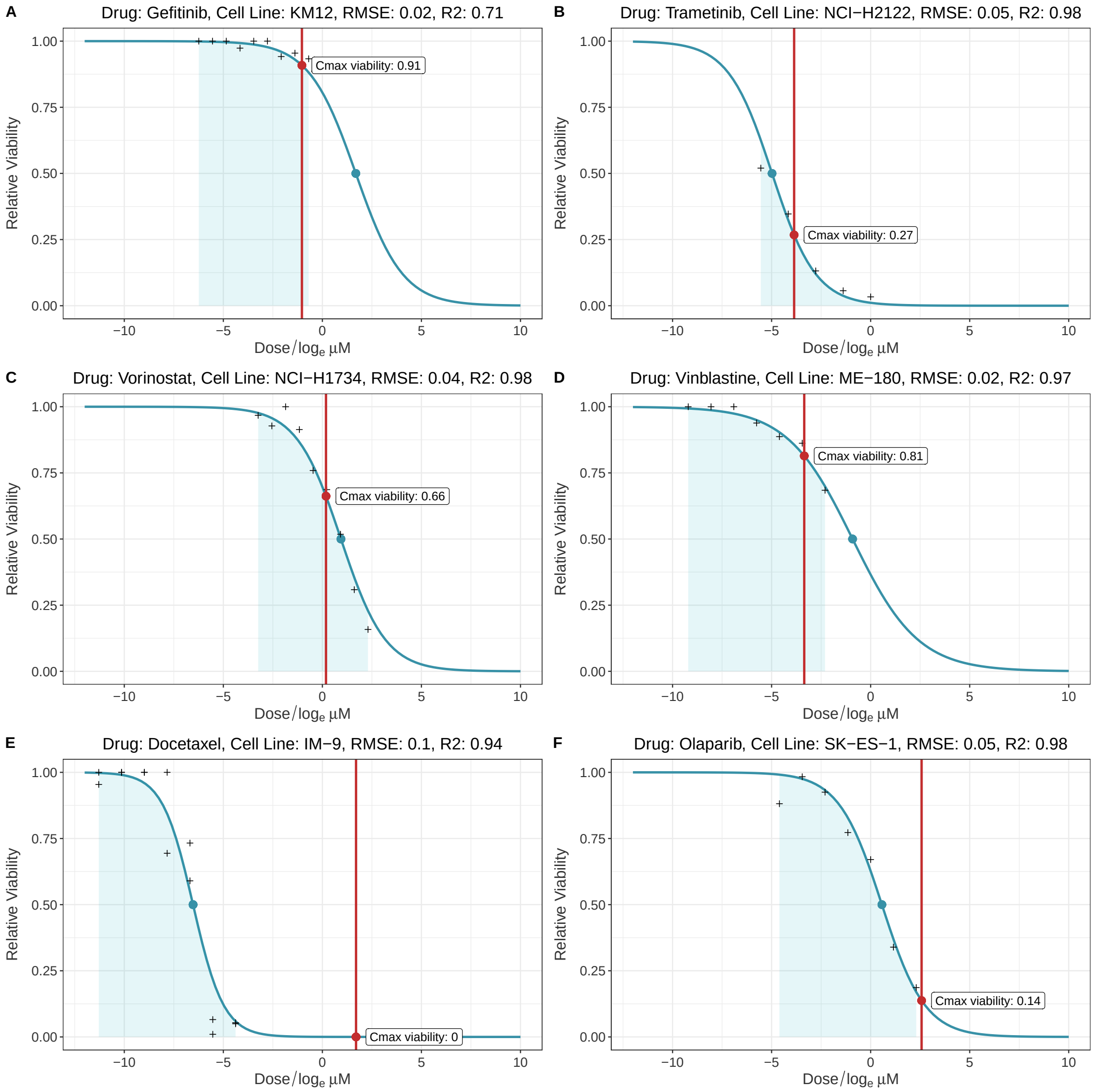


FIGURE E.1: Exemplary dose-response curves for six drug and cell line combinations. In each subfigure, the dose-response measurements as provided by the GDSC are shown as black crosses. The corresponding dose-response curve is shown in blue and the blue area below the curve denotes the tested concentration range. The thick blue dot marks the point where the curve reaches a relative viability of 0.5. The concentration at this point is the IC50 value. The red vertical line denotes the CMax concentration of the respective drug and the thick red dot marks the point where the curves reaches this concentration. The relative viability at this position is the CMax viability.

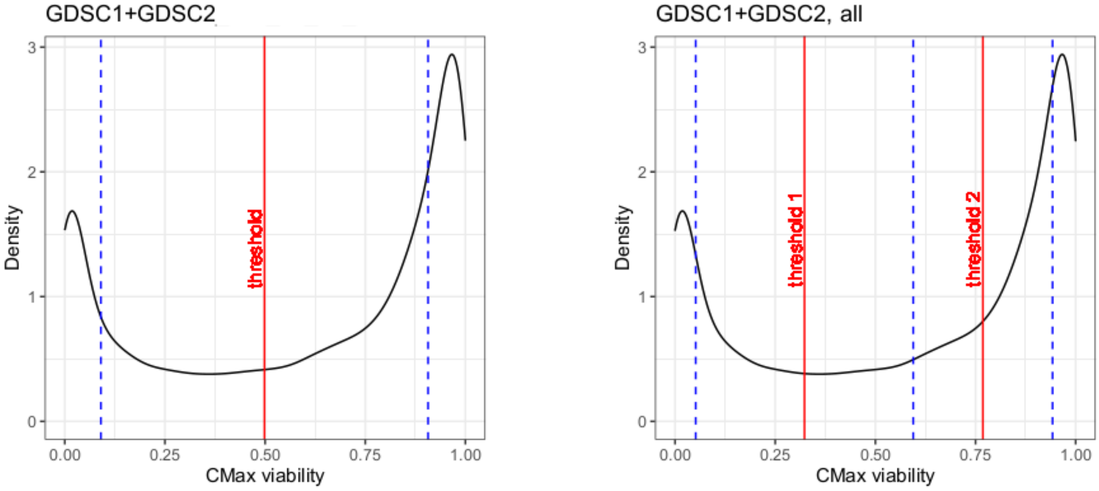


FIGURE E.2: Distribution of CMax viabilities. Both plots show the distribution of CMax viability values across all cell line-drug combinations where this measure could be computed. The left plot depicts the cluster medoids (blue dashed lines) and threshold ($t = 0.5$, red line) of a two-class PAM clustering (classes: *sensitive*, *resistant*). The right plot depicts the medoids and thresholds ($t_1 = 0.32$, $t_2 = 0.77$) of a three-class PAM clustering (classes: *sensitive*, *intermediate*, *resistant*).

E.1 Selection of Drugs for Drug-Centric Analyses

We only considered drugs where both IC50 and CMax viability values were available (60 from GDSC1, 47 from GDSC2). Furthermore, we discarded drugs with an insufficient number of samples per class. To this end, we considered an application case where we not only perform training, calibration, and testing but also a cross-validation (CV) on the training data, e.g., for hyperparameter tuning. To enable performing conformal prediction inside the cross-validation, we propose a 5-fold CV procedure where four folds are used for training and the fifth fold is divided in half for calibration and testing.

In this setting, we need at least 12 samples of each class to theoretically generate a partition of samples into training, calibration, and test sets (also inside of the CV) where each set contains at least one sample per class. Consequently, we discarded all drugs for which less than 12 samples per class were available and randomly sampled the three datasets from the available cell lines of each drug.

Note that the True Class and Summation score do not require each class to be present in the calibration data, while the Mondrian score does. We decided to discard drugs where this condition was not fulfilled instead of repeatedly resampling the datasets until the condition was fulfilled. In Figures E.3 to E.6, we visualize which drugs were included and discarded from the analyses.

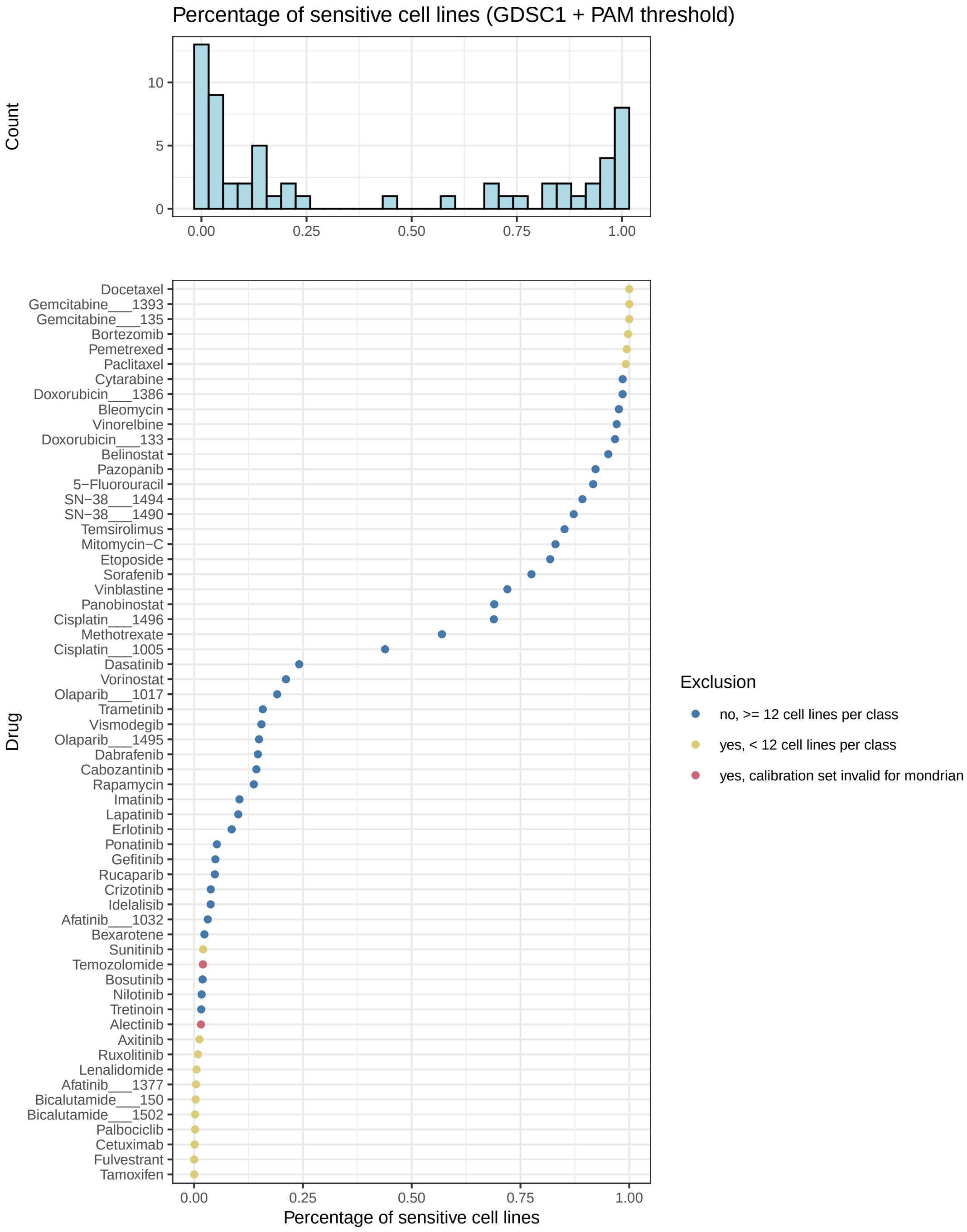


Figure E.3: Percentage of sensitive cell lines for binarized CMax viability in GDSC1. The lower part of this figure shows the percentage of sensitive cell lines for each drug when deriving discretized drug responses from a PAM clustering of CMax viabilities. The colors denote whether and why certain drugs were excluded from our analyses. The upper row shows a distribution of the percentage of sensitive cell lines across drugs.

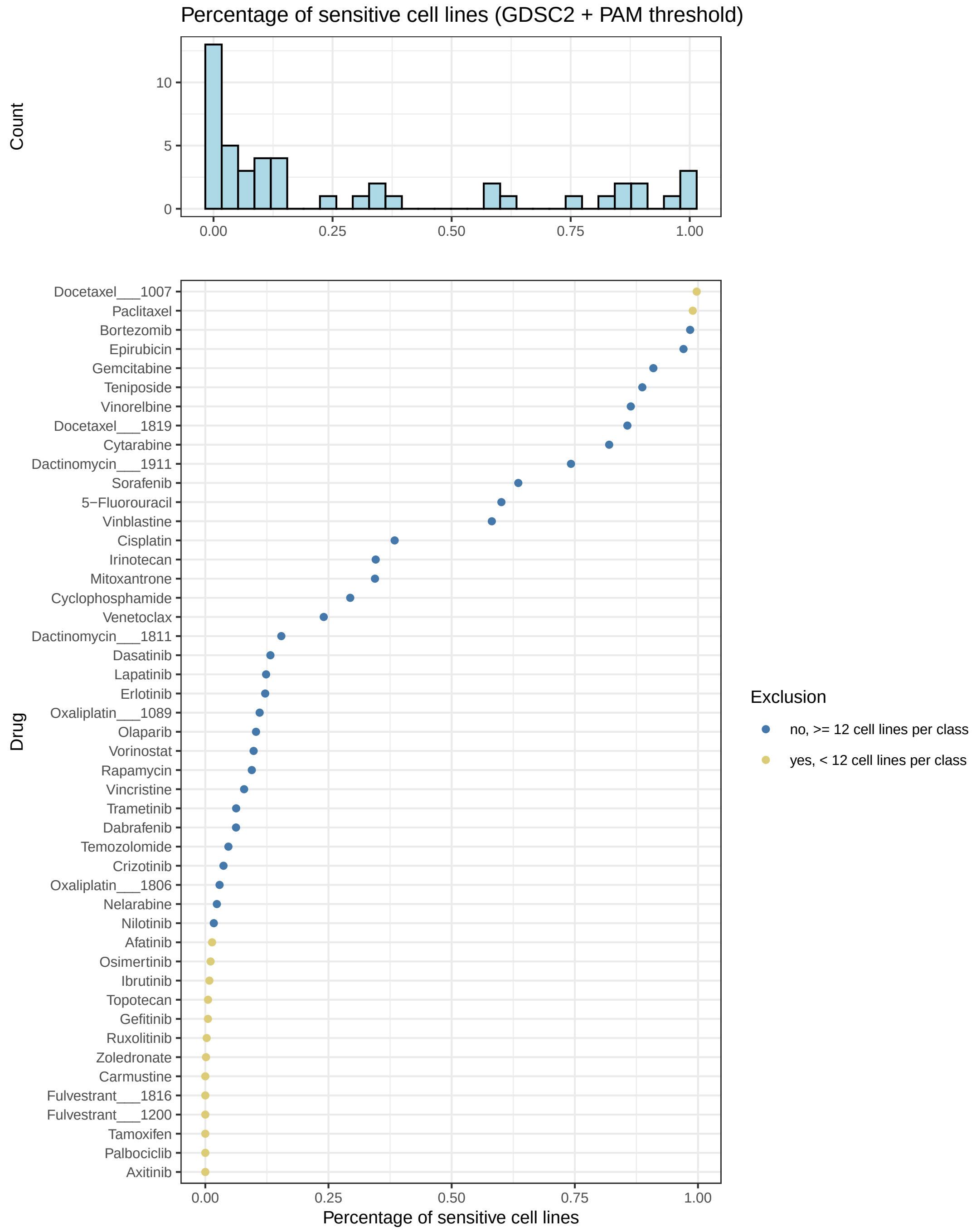


FIGURE E.4: Percentage of sensitive cell lines for binarized CMax viability in GDSC2. The lower part of this figure shows the percentage of sensitive cell lines for each drug when deriving discretized drug responses from a PAM clustering of CMax viabilities. The colors denote whether and why certain drugs were excluded from our analyses. The upper row shows a distribution of the percentage of sensitive cell lines across drugs.

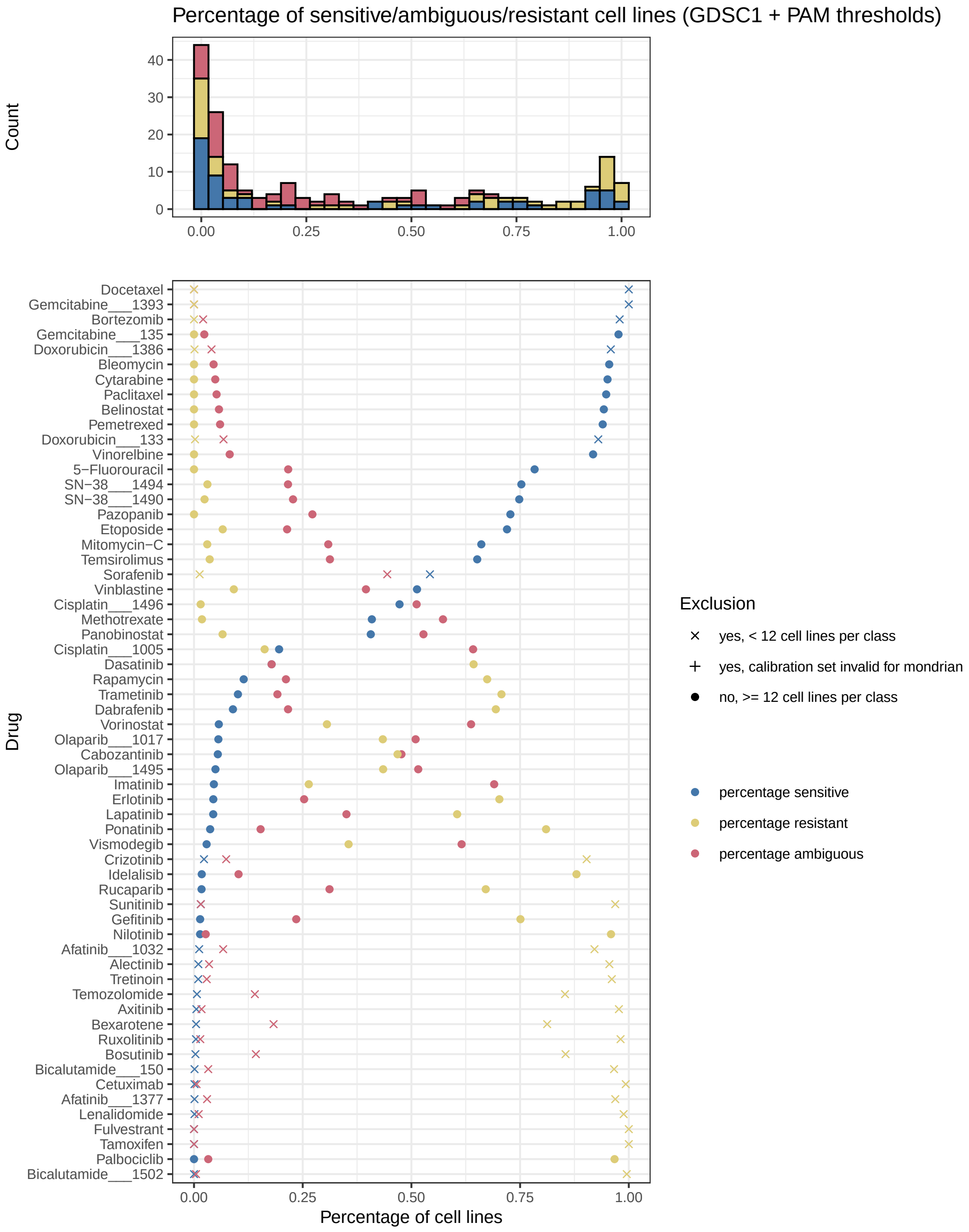


Figure E.5: Percentage of sensitive, intermediate, and resistant cell lines of ternary CMax viability in GDSC1. The lower part of this figure shows the percentages of sensitive, intermediate (here denoted as *ambiguous*) and resistant cell lines for each drug when deriving discretized drug responses from a PAM clustering of CMax viabilities. The shape of each point denotes whether and why certain drugs were excluded from our analyses. The upper row shows a class distribution across drugs.

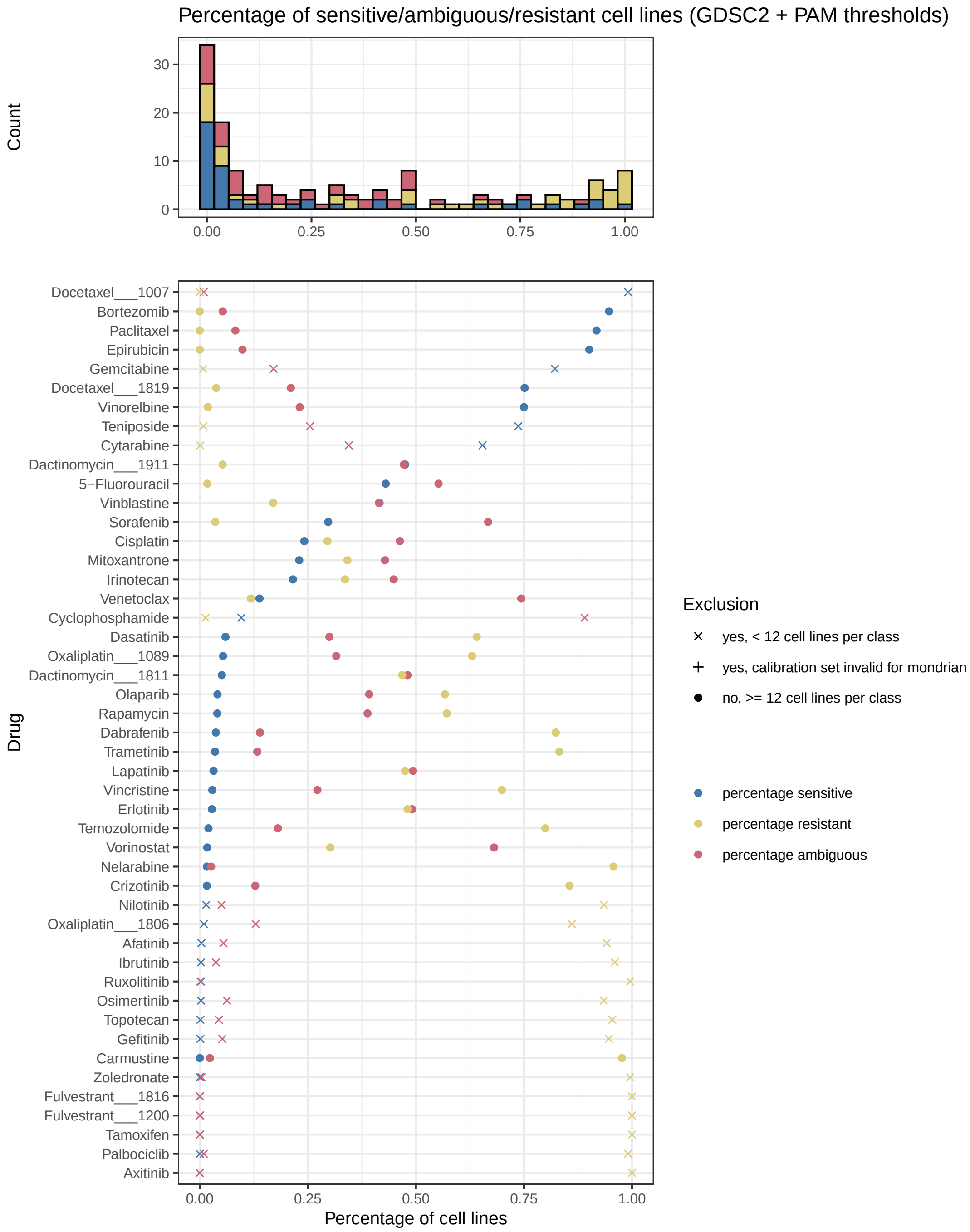


Figure E.6: Percentage of sensitive, intermediate, and resistant cell lines of ternary CMax viability in GDSC2. The lower part of this figure shows the percentages of sensitive, intermediate (here denoted as *ambiguous*) and resistant cell lines for each drug when deriving discretized drug responses from a PAM clustering of CMax viabilities. The shape of each point denotes whether and why certain drugs were excluded from our analyses. The upper row shows a class distribution across drugs.

Table E.1: Investigated drugs from GDSC1. This table contains all drugs we investigated from the GDSC1 dataset. For each drug, the number of cell lines is provided, and it is denoted whether the drug was included in the two-class classification, three-class classification, and prioritization analyses, respectively.

	Drug	Cell lines	Two classes	Three classes	Prioritization
1	SN-38___1494	952	yes	yes	yes
2	Vismodegib	949	yes	yes	yes
3	Cisplatin___1005	948	yes	yes	yes
4	Vorinostat	948	yes	yes	yes
5	Methotrexate	947	yes	yes	yes
6	Nilotinib	947	yes	yes	no
7	Olaparib___1017	946	yes	yes	yes
8	Bosutinib	946	yes	no	no
9	Vinblastine	945	yes	yes	yes
10	Afatinib___1032	943	yes	no	no
11	Gefitinib	941	yes	yes	no
12	Rucaparib	940	yes	yes	no
13	Cytarabine	937	yes	yes	no
14	Tretinoin	937	yes	no	no
15	Doxorubicin___1386	934	yes	no	no
16	Cisplatin___1496	931	yes	yes	yes
17	Temsirolimus	928	yes	yes	yes
18	SN-38___1490	924	yes	yes	yes
19	Idelalisib	920	yes	yes	no
20	Cabozantinib	917	yes	yes	yes
21	5-Fluorouracil	913	yes	yes	yes
22	Olaparib___1495	906	yes	yes	yes
23	Pemetrexed	905	no	yes	no
24	Panobinostat	904	yes	yes	yes
25	Palbociclib	901	no	yes	no
26	Belinostat	892	yes	yes	no
27	Trametinib	892	yes	yes	yes
28	Vinorelbine	887	yes	yes	no
29	Etoposide	886	yes	yes	yes
30	Ponatinib	884	yes	yes	no
31	Doxorubicin___133	877	yes	no	no
32	Mitomycin-C	877	yes	yes	yes
33	Pazopanib	873	yes	yes	yes
34	Dabrafenib	870	yes	yes	yes
35	Bexarotene	869	yes	no	no
36	Gemcitabine___135	865	no	yes	no
37	Bleomycin	862	yes	yes	no
38	Imatinib	407	yes	yes	yes
39	Crizotinib	406	yes	no	no
40	Sorafenib	398	yes	no	yes
41	Paclitaxel	397	no	yes	no
42	Lapatinib	396	yes	yes	yes
43	Erlotinib	394	yes	yes	yes
44	Dasatinib	394	yes	yes	yes
45	Rapamycin	358	yes	yes	yes

Table E.2: Investigated drugs from GDSC2. This table contains all drugs we investigated from the GDSC2 dataset. For each drug, the number of cell lines is provided, and it is denoted whether the drug was included in the two-class classification, three-class classification, and prioritization analyses, respectively.

	Drug	Cell lines	Two classes	Three classes	Prioritization
1	5-Fluorouracil	806	yes	yes	yes
2	Oxaliplatin___1089	802	yes	yes	yes
3	Irinotecan	801	yes	yes	yes
4	Trametinib	771	yes	yes	yes
5	Cisplatin	768	yes	yes	yes
6	Olaparib	762	yes	yes	yes
7	Dasatinib	760	yes	yes	yes
8	Vorinostat	758	yes	yes	yes
9	Nilotinib	757	yes	no	no
10	Paclitaxel	757	no	yes	no
11	Sorafenib	757	yes	yes	yes
12	Dabrafenib	757	yes	yes	yes
13	Lapatinib	757	yes	yes	yes
14	Gemcitabine	756	yes	no	yes
15	Bortezomib	756	yes	yes	no
16	Venetoclax	756	yes	yes	yes
17	Cytarabine	752	yes	no	yes
18	Erlotinib	752	yes	yes	yes
19	Epirubicin	752	yes	yes	no
20	Cyclophosphamide	751	yes	no	yes
21	Vinblastine	750	yes	yes	yes
22	Temozolomide	750	yes	yes	no
23	Vinorelbine	750	yes	yes	yes
24	Rapamycin	745	yes	yes	yes
25	Dactinomycin___1911	740	yes	yes	yes
26	Crizotinib	737	yes	yes	no
27	Oxaliplatin___1806	728	yes	no	no
28	Carmustine	728	no	yes	no
29	Teniposide	728	yes	no	yes
30	Mitoxantrone	728	yes	yes	yes
31	Dactinomycin___1811	728	yes	yes	yes
32	Nelarabine	727	yes	yes	no
33	Vincristine	722	yes	yes	yes
34	Docetaxel___1819	669	yes	yes	yes

E.2 Definition of Linear and Quadratic Sample Weights for Multi-Class Classification

To define the *linear* and *quadratic* sample weights, we assume that the K classes $C = \{c_1, ..., c_K\}$ are ordered ascendingly based on the thresholds that divide two neighboring classes. Let t_j denote the threshold that divides the classes c_j and c_{j+1} for all $j \in \{1, ..., K-1\}$ with $t_1 < ... < t_{K-1}$. The weight for sample x_i with discrete response $d_i \in C$ and continuous response y_i is then calculated as:

$$w_i^* = \begin{cases} \frac{|y_i - t_1|^g}{K \cdot \sum_{\forall n \in \{1,...,N\}: d_n = d_i} |y_n - t_1|^g}, & \text{if } d_i = c_1 \\ \frac{|y_i - t_{K-1}|^g}{K \cdot \sum_{\forall n \in \{1,...,N\}: d_n = d_i} |y_n - t_{K-1}|^g}, & \text{if } d_i = c_K \\ \frac{|y_i - t_{j-1}|^g + |y_i - t_j|^g}{K \cdot \sum_{\forall n \in \{1,...,N\}: d_n = d_i} |y_n - t_{j-1}|^g + |y_n - t_j|^g}, & \text{else, i.e., } d_i = c_j, j \in \{2, ..., K-1\} \end{cases} \tag{E.1}$$

Here, N denotes the total number of samples. For linear sample weights, $g = 1$, and for quadratic sample weights, $g = 2$. For the classes c_1 and c_K, the weights are equal to those presented in Equation 6.3 (where $K = 2$) since these classes only have one neighboring threshold. For all other classes, the distance of y_i from both neighboring thresholds is averaged.

E.3 Specifications Regarding Classification Performance

In the figures depicting the classification performance in Chapter 7 and in this appendix, we measure the classification performance in percentages. Here, we provide the explicit formulas how these percentages were calculated for the two-class classification. They can analogously be computed for the three-class classification.

$$\text{percent TP, case 1:1 (equal to sensitivity)} = \frac{TP}{TP+FN}$$

$$\text{percent FN, case 1:0 (equal to miss-rate)} = \frac{FN}{TP+FN}$$

$$\text{percent TN, case 0:0 (equal to specificity)} = \frac{TN}{TN+FP}$$

$$\text{percent FP, case 0:1 (equal to fall-out)} = \frac{FP}{TN+FP}$$

$$\text{case } 1:[] = \frac{\text{\#sensitive samples with empty class prediction}}{TP+FN}$$

$$\text{case } 1:[1,0] = \frac{\text{\#sensitive samples with [1,0] as class prediction}}{TP+FN}$$

$$\text{case } 1:[0,1] = \frac{\text{\#sensitive samples with [0,1] as class prediction}}{TP+FN}$$

$$\text{case } 0:[] = \frac{\text{\#resistant samples with empty class prediction}}{TN+FP}$$

$$\text{case } 0:[1,0] = \frac{\text{\#resistant samples with [1,0] as class prediction}}{TN+FP}$$

$$\text{case } 0:[0,1] = \frac{\text{\#resistant samples with [0,1] as class prediction}}{TN+FP}$$

E.4 Analyses Using a Multi-Omics, Multi-Drug Deep Learning Approach by Chiu et al.

We wanted to investigate the impact of CP on a state-of-the-art method for drug sensitivity prediction other than SAURON-RF and to compare the performance of this method to SAURON-RF. Since the reliability guarantees and drug prioritization pipeline introduced in Chapter 7 have not been described in any previous publications (cf. Table 7.1), we cannot compare our approach to any existing methods directly. Consequently, we performed several analyses using a slightly modified version of the multi-omics, multi-drug deep learning approach by Chiu et al. [292]. We already utilized this approach for some analyses in Chapter 5, the details of which are provided in Appendix C.2. In the following, we will again briefly present their approach and then describe the details of our analyses.

E.4.1 The Approach by Chiu et al.

Chiu et al. developed a multi-omics deep neural network (DNN) for drug sensitivity prediction that predicts the IC50 of multiple drugs simultaneously. The inputs consist of gene expression values and binary mutation data for one cell line. Using one expression-autoencoder and one mutation-autoencoder, these inputs are projected into a lower dimension of $k = 64$ features each. The autoencoders were pre-trained using data from tumor samples obtained from *The Cancer Genome Atlas* (TCGA, `https://www.cancer.gov/tcga`). The pre-trained encoders are then connected to a DNN with drug-specific output nodes. The entire model was trained and evaluated using cell line data from the *Cancer Cell Line Encyclopedia* (CCLE) [484].

E.4.2 Classification with CP Based on Discretized IC50 Values

In a first analysis, we investigated the classification performance of the DNN by Chiu et al. without and with CP. The results of this analysis are discussed in Chapter 7.2.1.2 and shown in Appendix Figures E.7 and E.8.

E.4.2.1 Data Processing

To apply the approach by Chiu et al. to the GDSC data and to compare its performance with and without CP to SAURON-RF, we prepared the data as follows:

- *Gene expression data:* We employ the same gene expression data as described in Chapter 7.
- *Mutation data:* We generated a binary mutation matrix $M_{cells \times genes}$, where each entry $M_{c,g}$ denotes whether gene g is mutated in cell line c ($M_{c,g} = 1$) or not ($M_{c,g} = 0$). We obtained coding point mutations of the GDSC cell lines from v99 of the COSMIC cell line project (file: CellLinesProject_GenomeScreensMutant_v99_GRCh37.tsv). In accordance with Chiu et al., we did not consider synonymous mutations.
- *Drug response data:* As responses, we use the discretized IC50 values as described in Chapter 7. However, since the model by Chiu et al. makes predictions for multiple drugs simultaneously, it requires data where each investigated cell line provides a drug response for each investigated drug. In the GDSC, not all cell lines have been screened against all drugs. To determine a maximal but complete subset of cell lines and drugs for our analyses, we applied an integer linear program (ILP) that can be found in Appendix C.3. This ILP determined a set consisting of 600 cell lines and 170 drugs.
- *Splitting into training, calibration, and test data:* In line with the analyses in Chapter 7, we randomly split the 600 cell lines into a training set (70% of cell lines), calibration set (15%), and test set (15%).

E.4.2.2 Model Architecture and Hyperparameters

- Where possible, we used the same model architecture and hyperparameters that Chiu et al. employ in their code (`https://github.com/chenlabgccri/DeepDR`). However, since the approach by Chiu et al. is only designed for regression, we made it suitable for classification (with CP) through the following modifications: We replaced the linear activation function in the last network layer with a sigmoid function. Consequently, the model predictions can be interpreted as the probability of a sample being sensitive towards each drug. The probability of a sample being resistant can thus be derived as 1 minus the predicted probability. Additionally, we replaced the MSE loss function with the binary cross-entropy (cf. Equation 4.64).
- Both the CCLE and TCGA data employed by Chiu et al. measure gene expression using RNA-seq, while the GDSC used for our analyses relies on microarrays. Consequently, pre-training using TCGA data was not possible for our analyses, so we used the training samples for pre-training instead. According to Chiu et al., TCGA

pre-trained models resulted in the best performance, but even using randomly initialized encoders outperformed all comparable analyses without pre-training [292]. Consequently, pre-training using the training cell lines should outperform most comparable alternatives including random initialization, when TCGA pre-training is not possible.

E.4.3 Regression Based on CMax Viabilities

In a second analysis, we investigated the regression performance of the DNN by Chiu et al. without CP. Unfortunately, there is no straightforward way to modify the DNN to allow using CP for regression. The results of this analysis are discussed in Chapter 7.2.1.2 and shown in Appendix Figures E.9 and E.10.

E.4.3.1 Data Processing

We used the same gene expression and mutation data as for the classification analyses presented above. As drug response, we employ the CMax viability, which is only available for 47 GDSC2 drugs. We intersected these 47 drugs with the 170 drugs determined through the ILP (cf. Section E.4.2.1) yielding 42 drugs for which we performed the regression analyses. Since no CP could be performed for regression, no calibration data is needed and we simply split the data into a training set (70 % of cell lines) and test set (30 %).

E.4.3.2 Model Architecture and Hyperparameters

We used the same architecture and hyperparameters as for the classification analyses presented above. However, since we perform regression here, we kept Chiu et al.'s original loss function (MSE) and activation function in the last network layer (linear function).

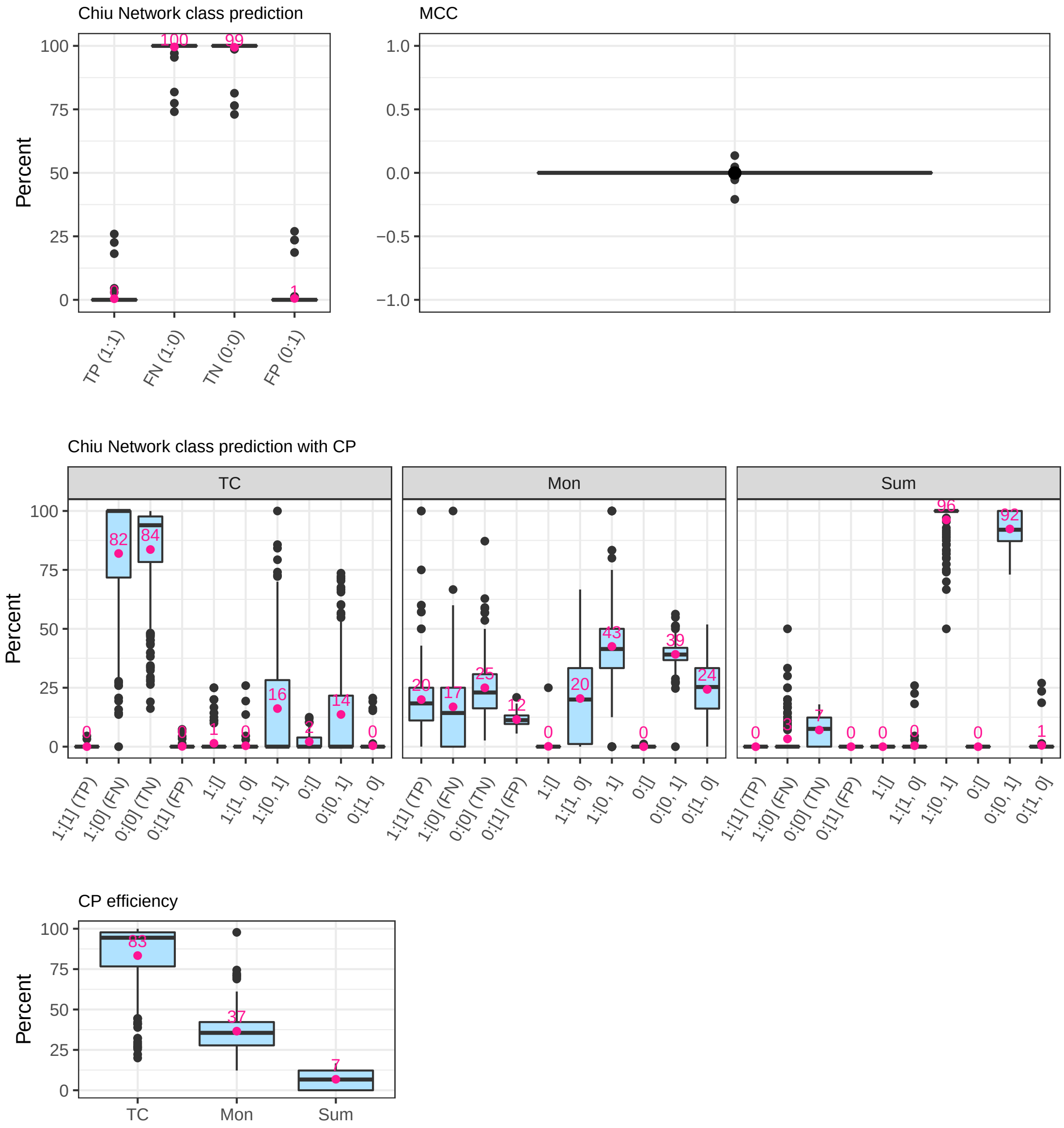


Figure E.7: Classification results of the DNN for 170 GDSC2 drugs in the two-class classification setting using IC50 values. This figure depicts the test set performance of the DNN without CP and with CP. The top row depicts the performance without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC across drugs. The middle row shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

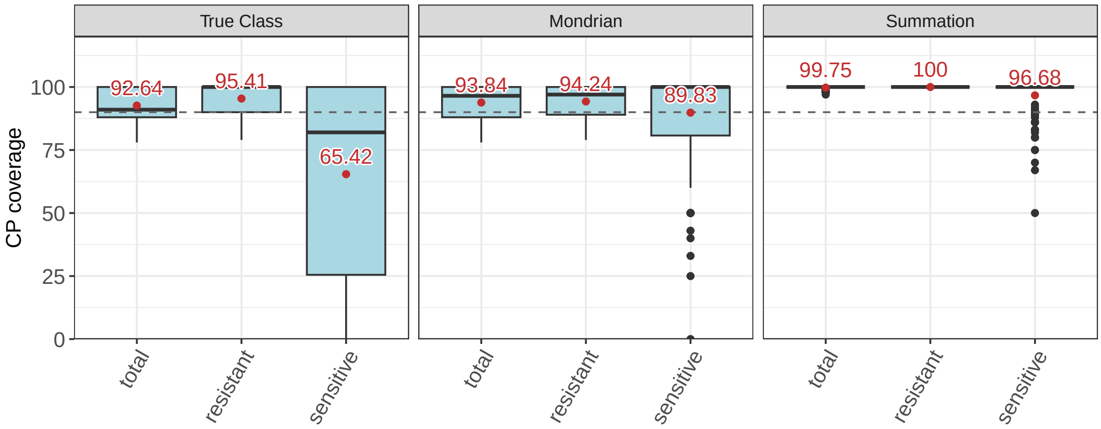


FIGURE E.8: Coverage evaluation of the DNN for 170 GDSC2 drugs in the two-class classification setting using IC50 values. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

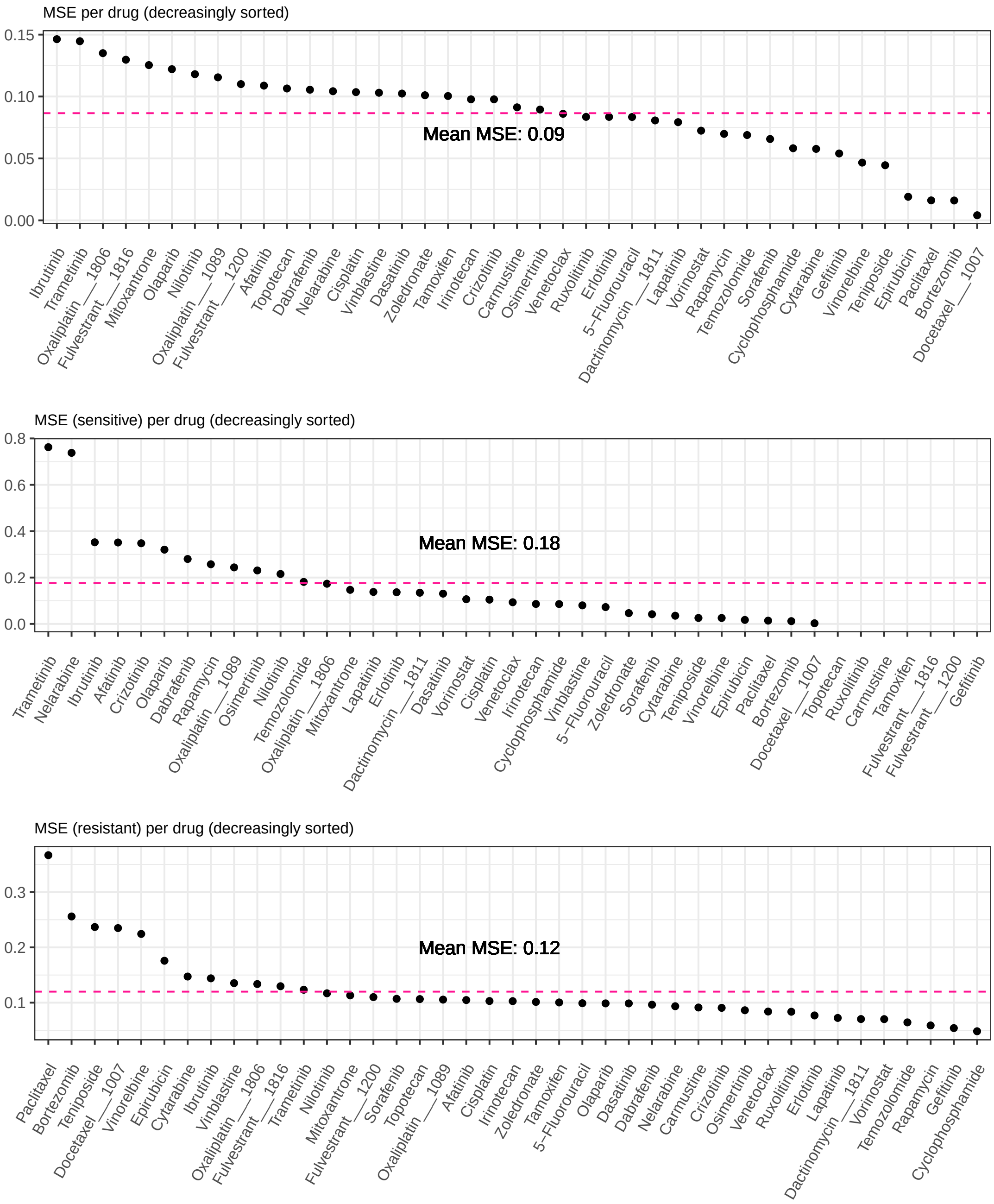


Figure E.9: MSEs of the DNN for 42 GDSC2 drugs in the regression setting using CMax viabilities. For each investigated drug, the overall test MSE (upper plot), as well as the test MSE for the subsets of sensitive (middle plot) and resistant cell lines (lower plot) are shown.

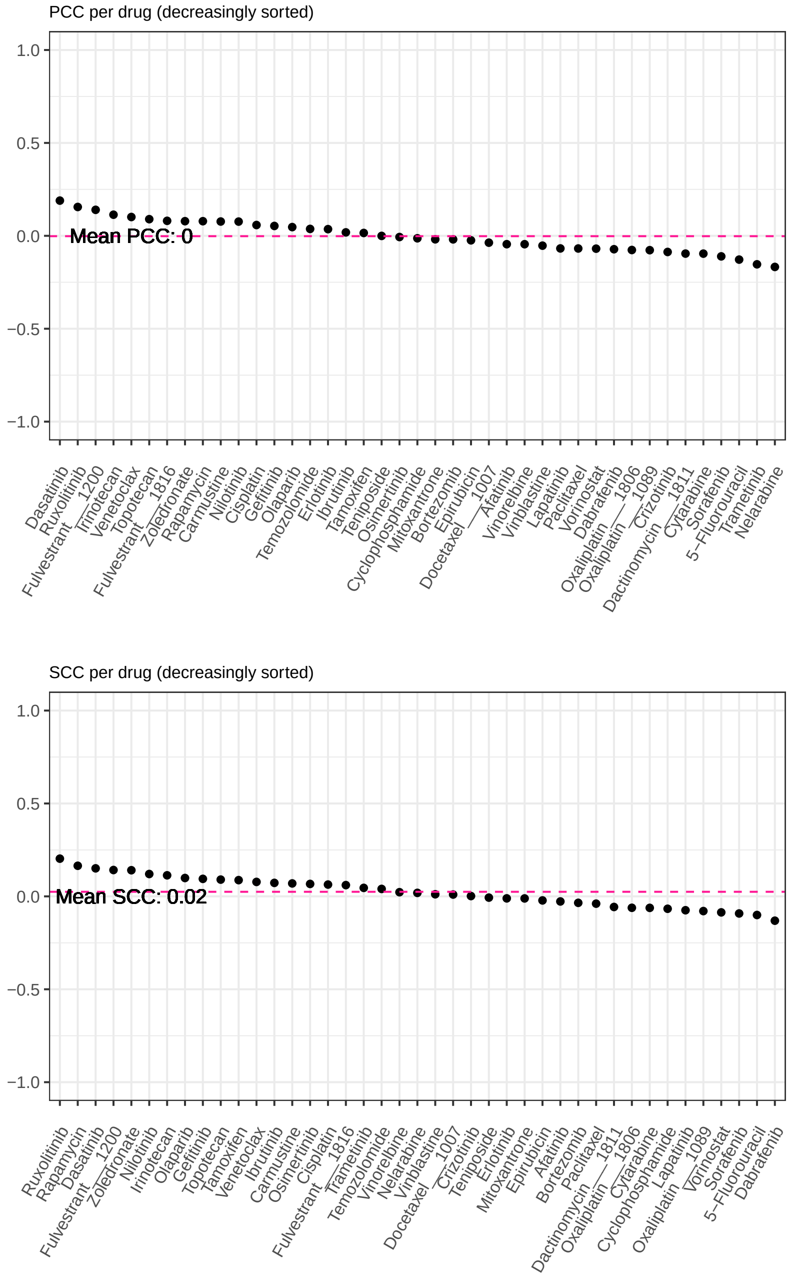


Figure E.10: PCC and SCC of the DNN for 42 GDSC2 drugs in the regression setting using CMax viabilities. For each investigated drug, the test set PCC (upper plot) and SCC (lower plot) are shown.

E.5 Additional Results for GDSC2 Database

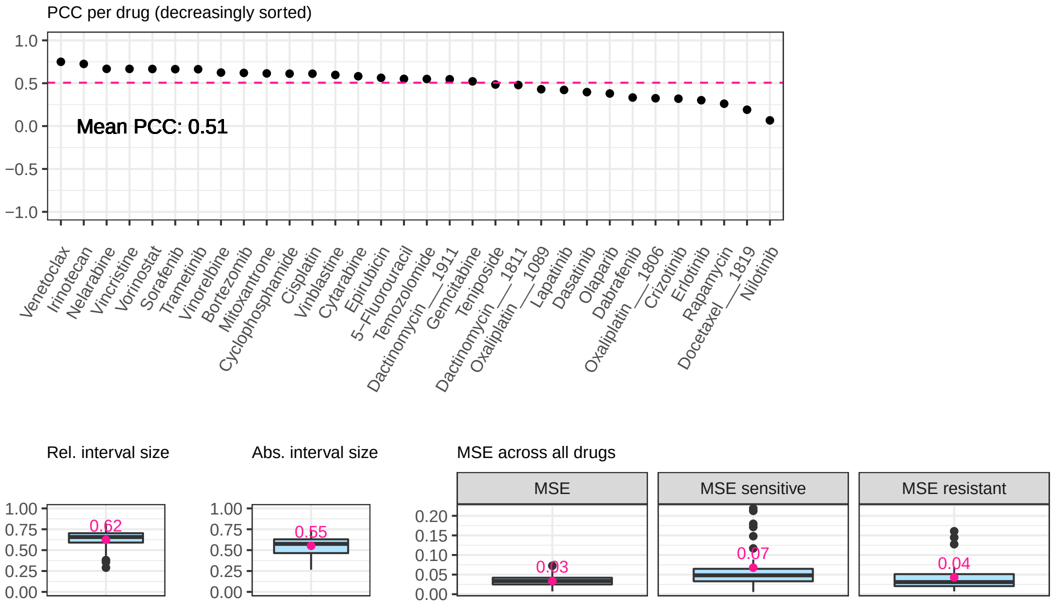


Figure E.11: Regression results for 32 GDSC2 drugs in the two-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive and resistant samples are shown.

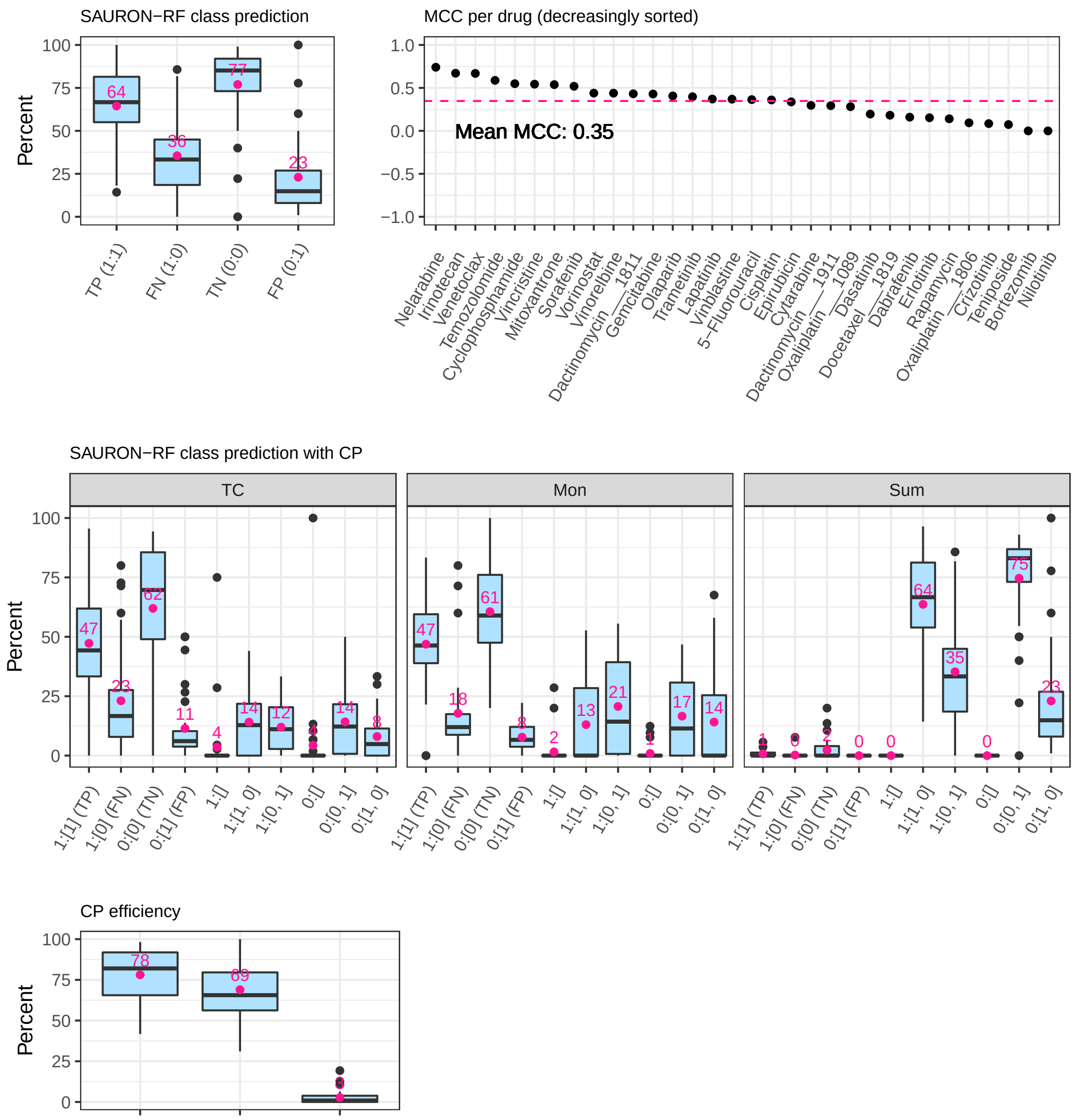


FIGURE E.12: Classification results for 32 GDSC2 drugs in the two-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the results without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC for each drug. The middle row shows how predictions are affected by applying CP with different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

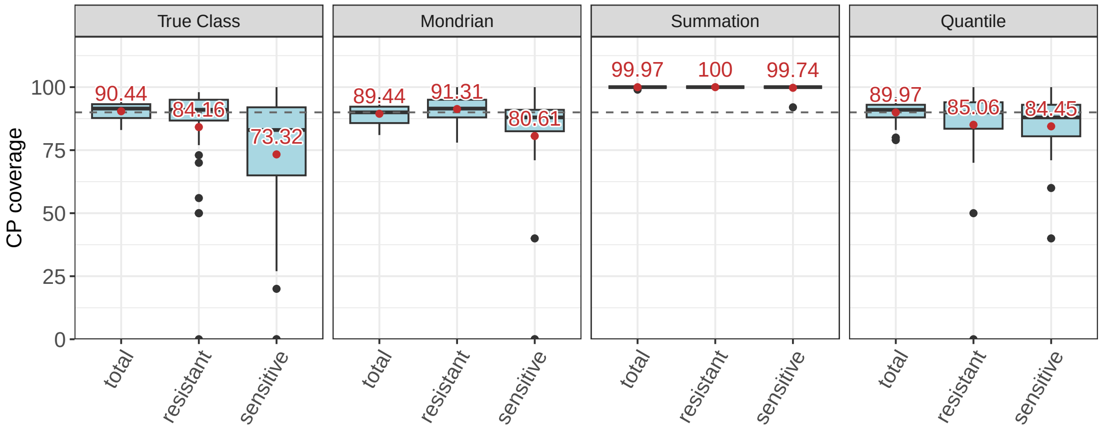


Figure E.13: Coverage evaluation for 32 GDSC2 drugs in the two-class classification setting using CMax viabilities. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

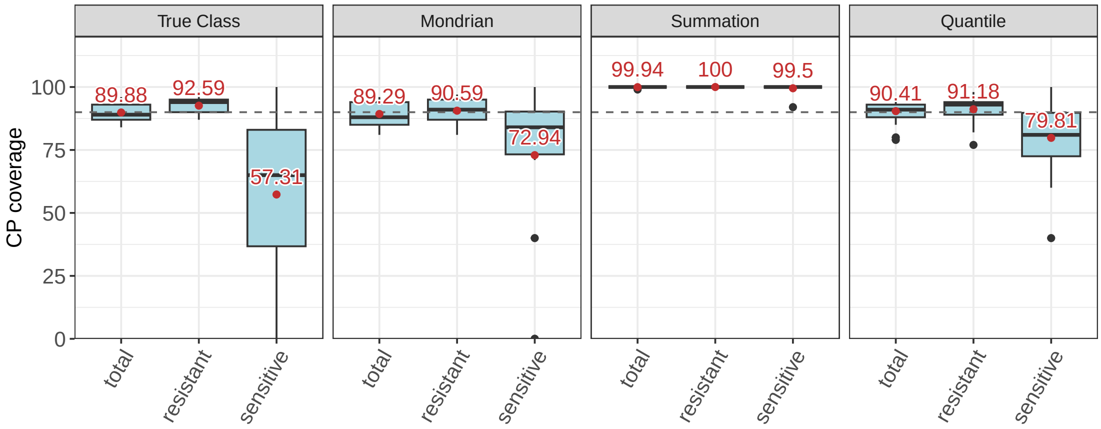


Figure E.14: Coverage evaluation for 17 frequently ineffective GDSC2 drugs in the two-class classification setting using CMax viabilities. Here, we only display those 17 drugs, for which ≥ 75% of samples are resistant. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

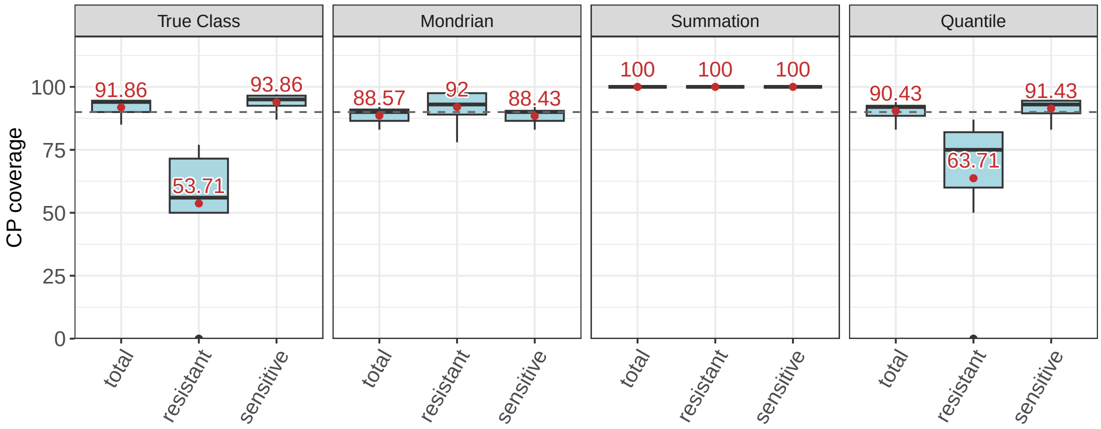


Figure E.15: Coverage evaluation for 7 frequently effective GDSC2 drugs in the two-class classification setting using CMax viabilities. Here, we only display those 7 drugs, for which ≥ 75% of samples are sensitive. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

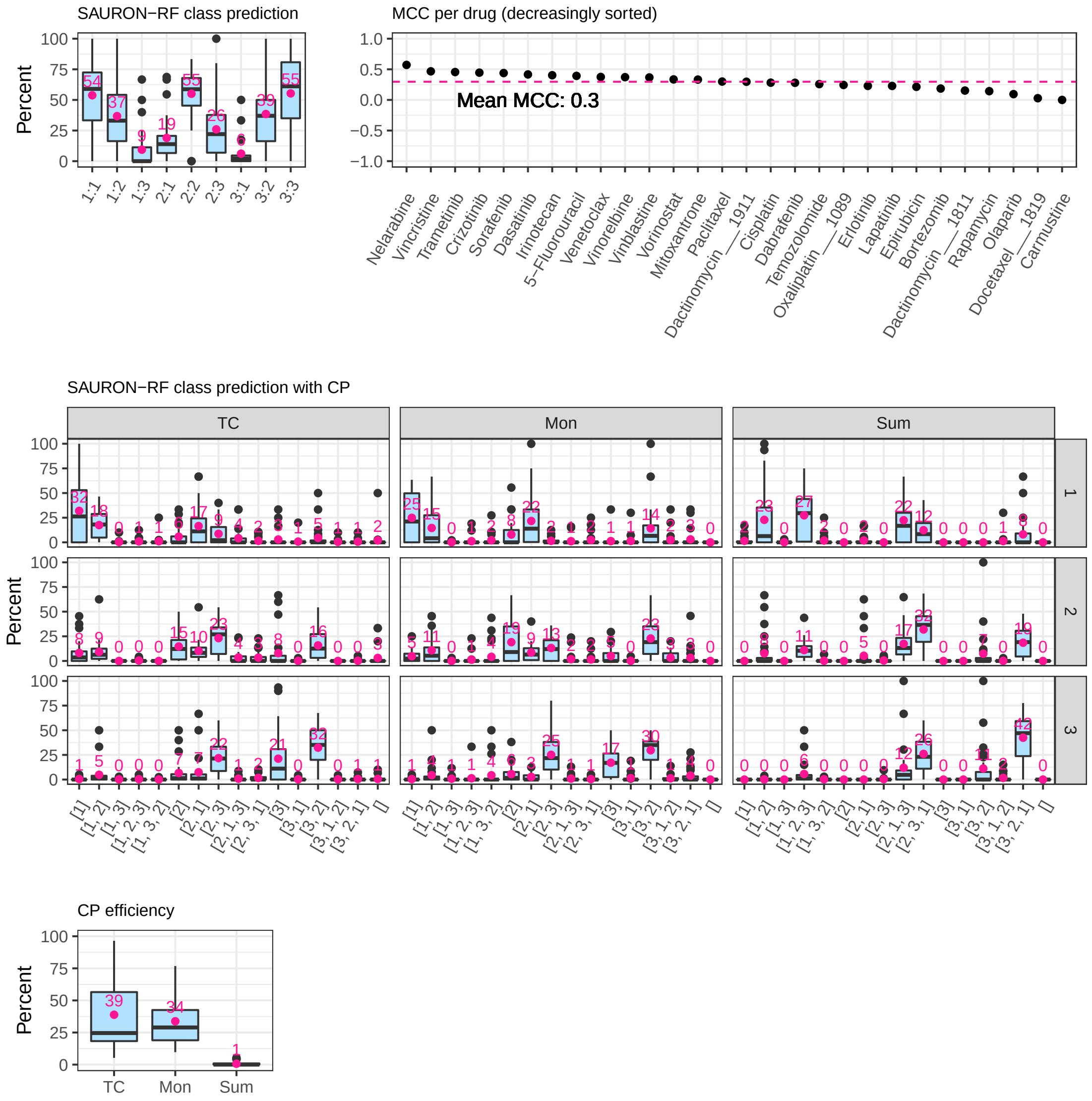


Figure E.16: Classification results for 28 GDSC2 drugs in the three-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of predictions across drugs as well as the MCC for each drug. In the left plot, each x-axis label denotes the actual and predicted class separated by a colon (1: sensitive, 2: intermediate, 3: resistant). The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). The middle row shows how predictions are affected by applying CP with three different classification scores. The actual class is shown on the right and the predicted sets are shown at the bottom. For two- and three-class sets, classes with higher prediction probability are listed first. In the bottom row, the CP efficiency for each score is shown.

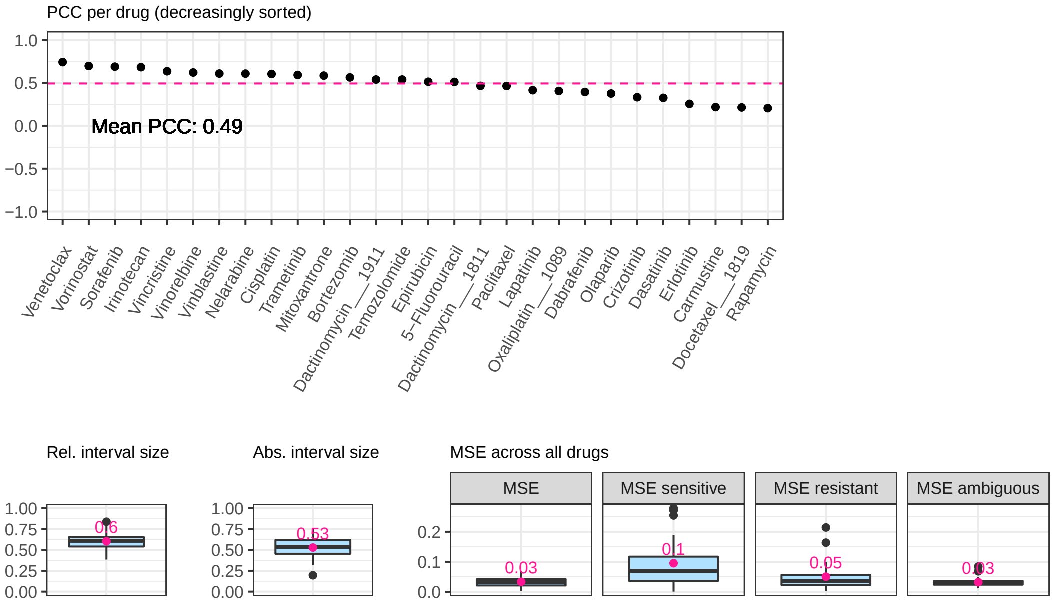


FIGURE E.17: Regression results for 28 GDSC2 drugs in the three-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive, intermediate (ambiguous), and resistant samples are shown.

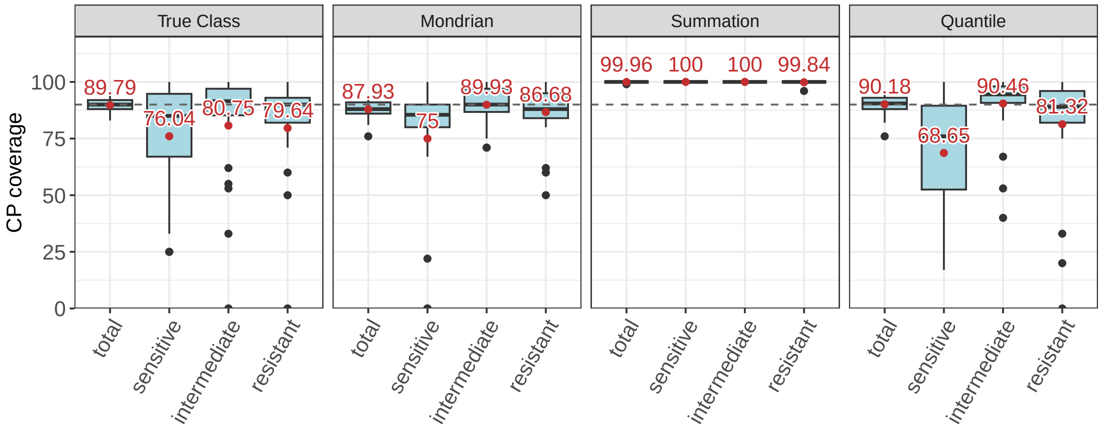


FIGURE E.18: Coverage evaluation for 28 GDSC2 drugs in the three-class classification setting using CMax viabilities. For each CP score, the total coverage as well as the coverage for the subsets of sensitive, intermediate, and resistant cell lines are shown.

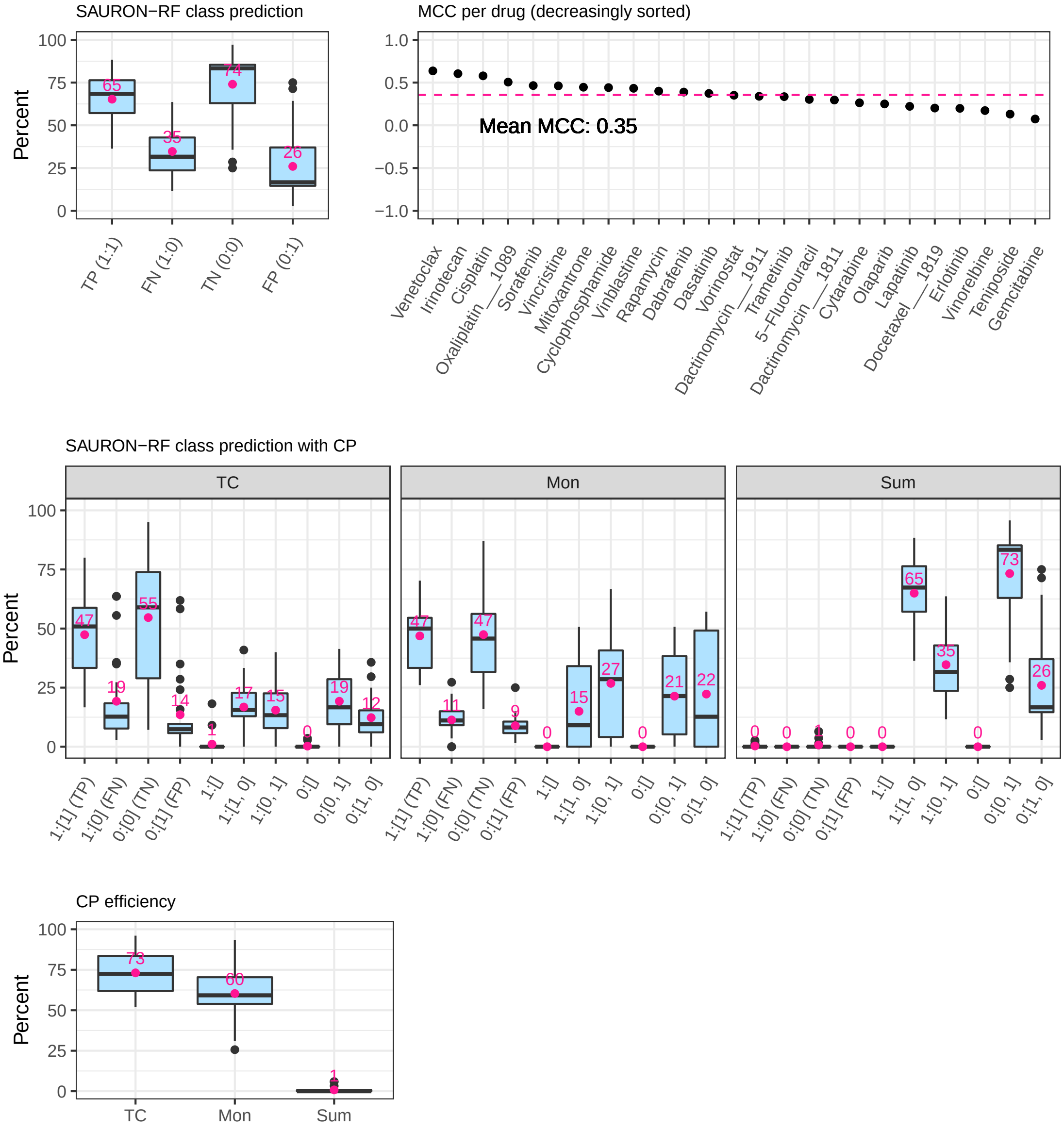


Figure E.19: Classification results for 25 GDSC2 drugs in the two-class prioritization setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC for each drug. The middle row shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

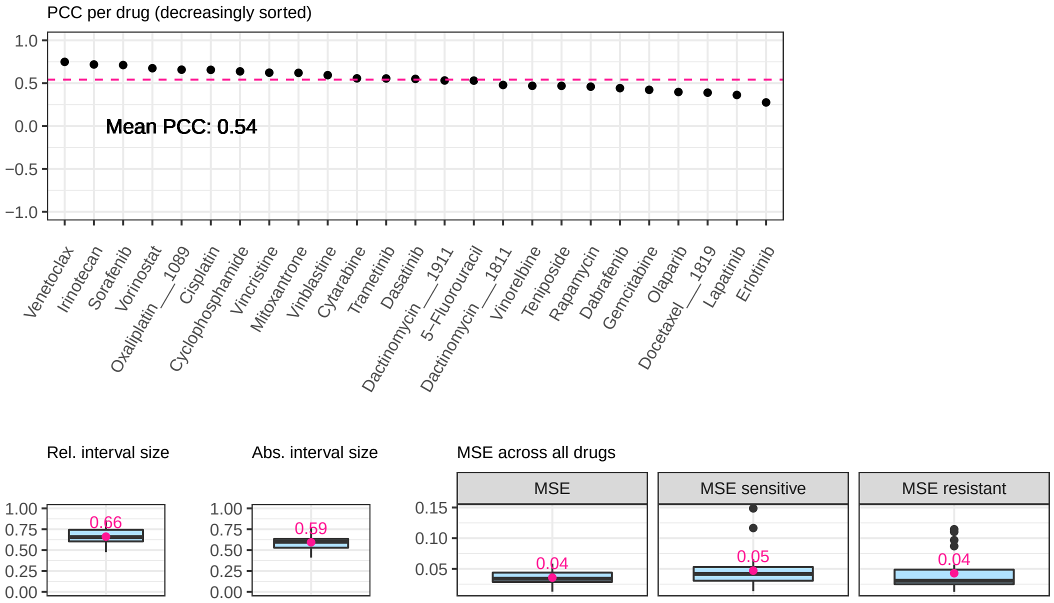


Figure E.20: Regression results for 25 GDSC2 drugs in the two-class prioritization setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive and resistant samples are shown.

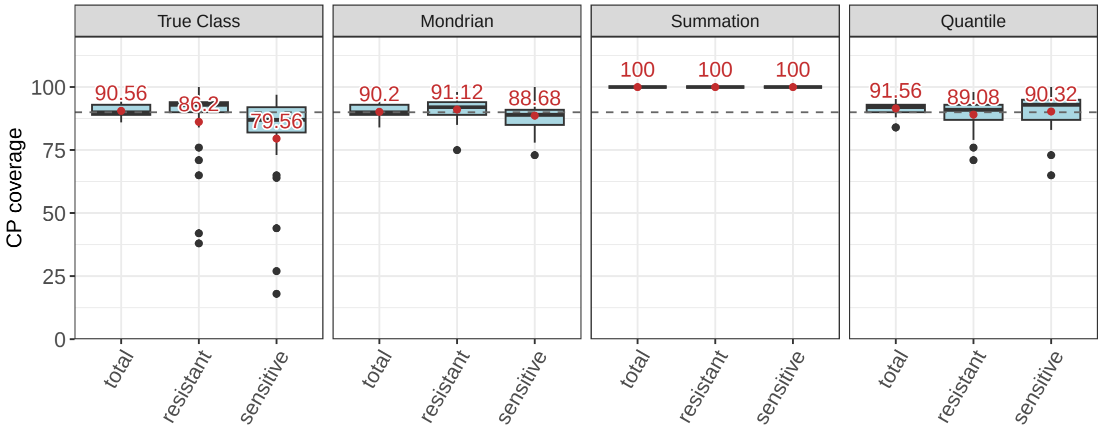


Figure E.21: Coverage evaluation for 25 GDSC2 drugs in the two-class prioritization setting using CMax viabilities. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

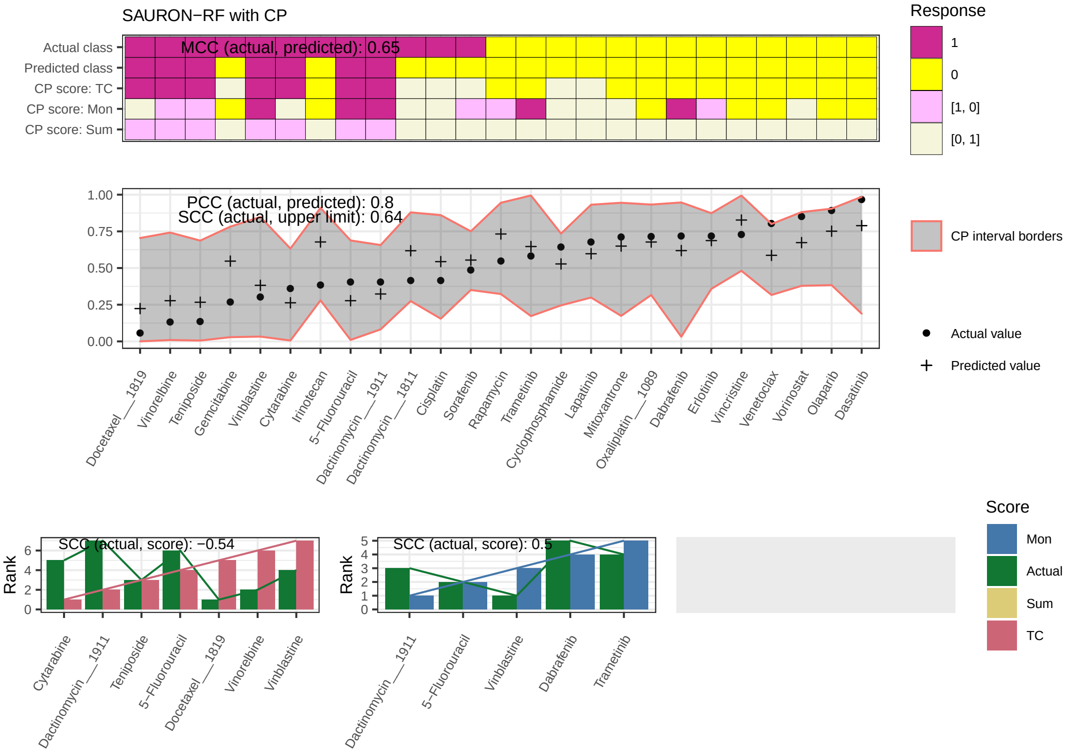


FIGURE E.22: Drug prioritization for the cell line with COSMIC-ID 724863 using GDSC2 drugs. The top and middle rows show the classification and regression results with and without CP. The bottom row shows the prioritized drug lists generated from the CP results: For each score, drugs predicted to cause a sensitive response are sorted ascendingly by the upper limit of the predicted response interval. Note that no results for the Summation score are shown since no drug was predicted to result in a sensitive response using this score.

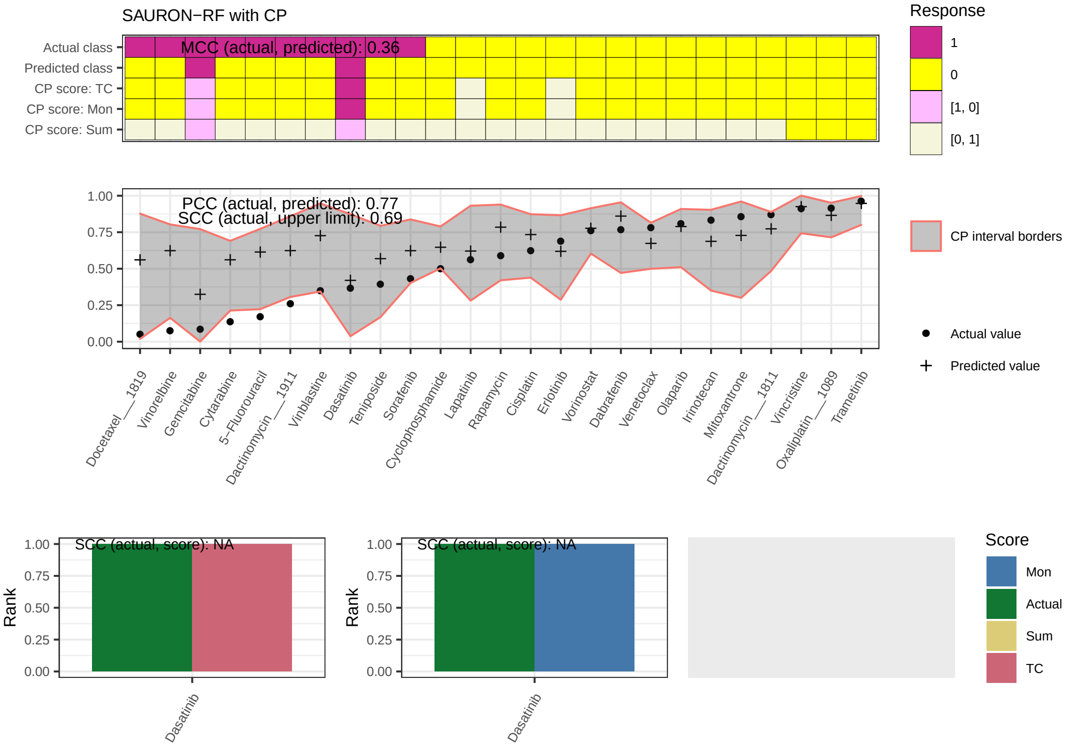


Figure E.23: Drug prioritization for the cell line with COSMIC-ID 906861 using GDSC2 drugs. The top and middle rows show the classification and regression results with and without CP. The bottom row shows the prioritized drug lists generated from the CP results: For each score, drugs predicted to cause a sensitive response are sorted ascendingly by the upper limit of the predicted response interval. Note that no results for the Summation score are shown since no drug was predicted to result in a sensitive response using this score.

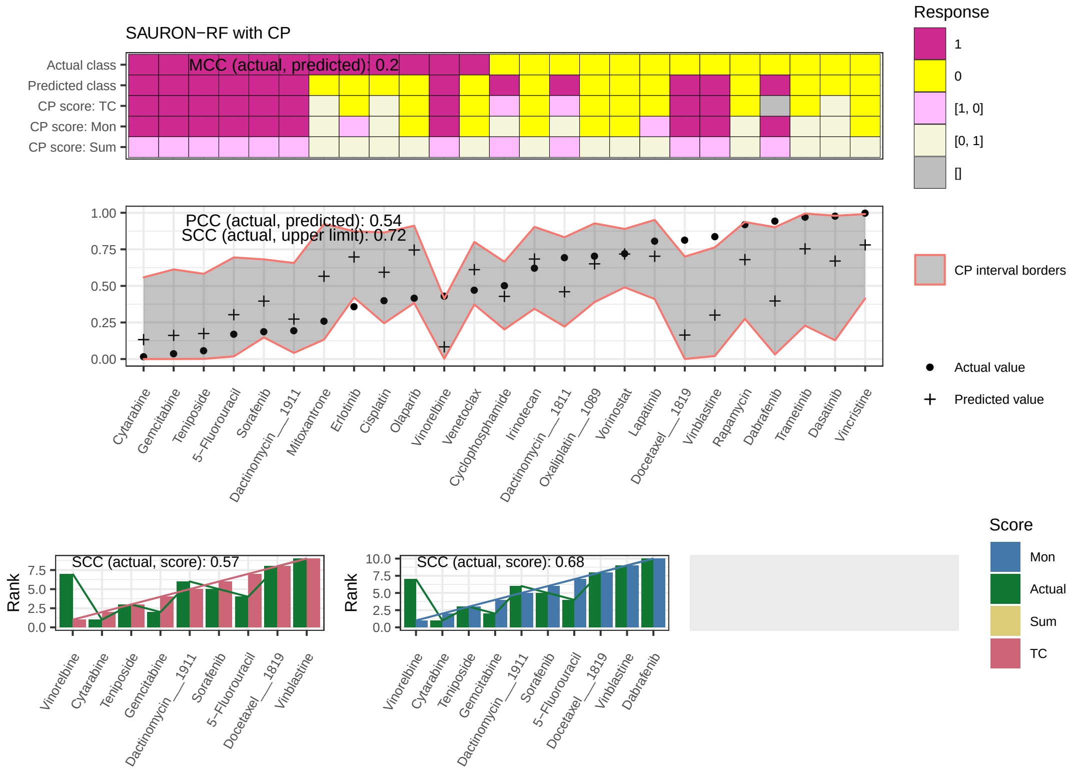


Figure E.24: Drug prioritization for the cell line with COSMIC-ID 908444 using GDSC2 drugs. The top and middle rows show the classification and regression results with and without CP. The bottom row shows the prioritized drug lists generated from the CP results: For each score, drugs predicted to cause a sensitive response are sorted ascendingly by the upper limit of the predicted response interval. Note that no results for the Summation score are shown since no drug was predicted to result in a sensitive response using this score.

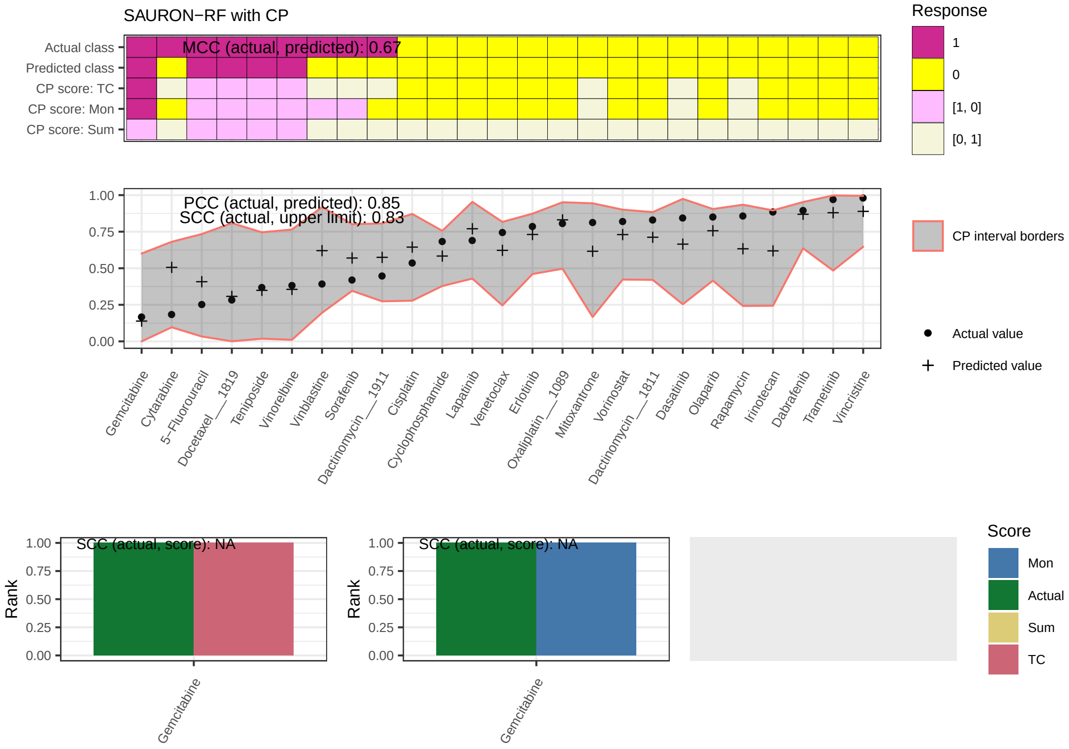


Figure E.25: Drug prioritization for the cell line with COSMIC-ID 1298216 using GDSC2 drugs. The top and middle rows show the classification and regression results with and without CP. The bottom row shows the prioritized drug lists generated from the CP results: For each score, drugs predicted to cause a sensitive response are sorted ascendingly by the upper limit of the predicted response interval. Note that no results for the Summation score are shown since no drug was predicted to result in a sensitive response using this score.

E.6 Results for GDSC1 Database

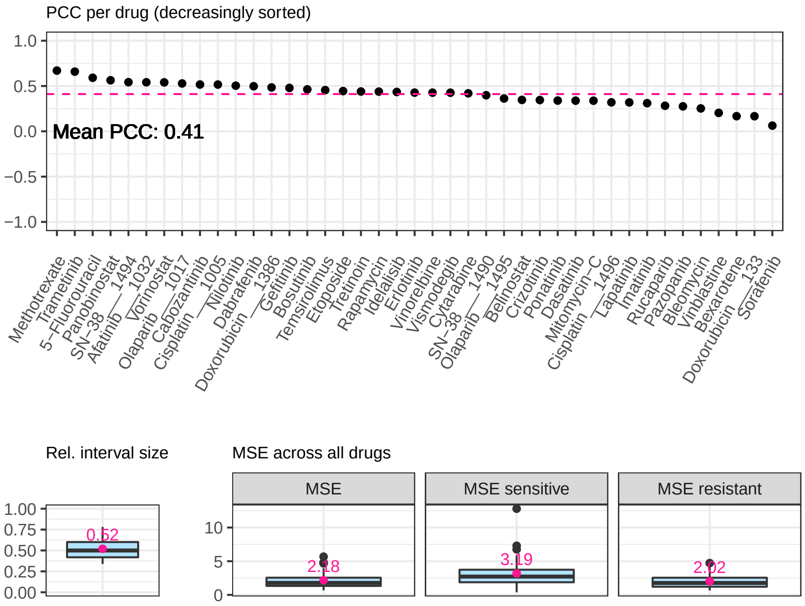


Figure E.26: Regression results for 41 GDSC1 drugs in the two-class classification setting using IC50 values. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive and resistant samples are shown.

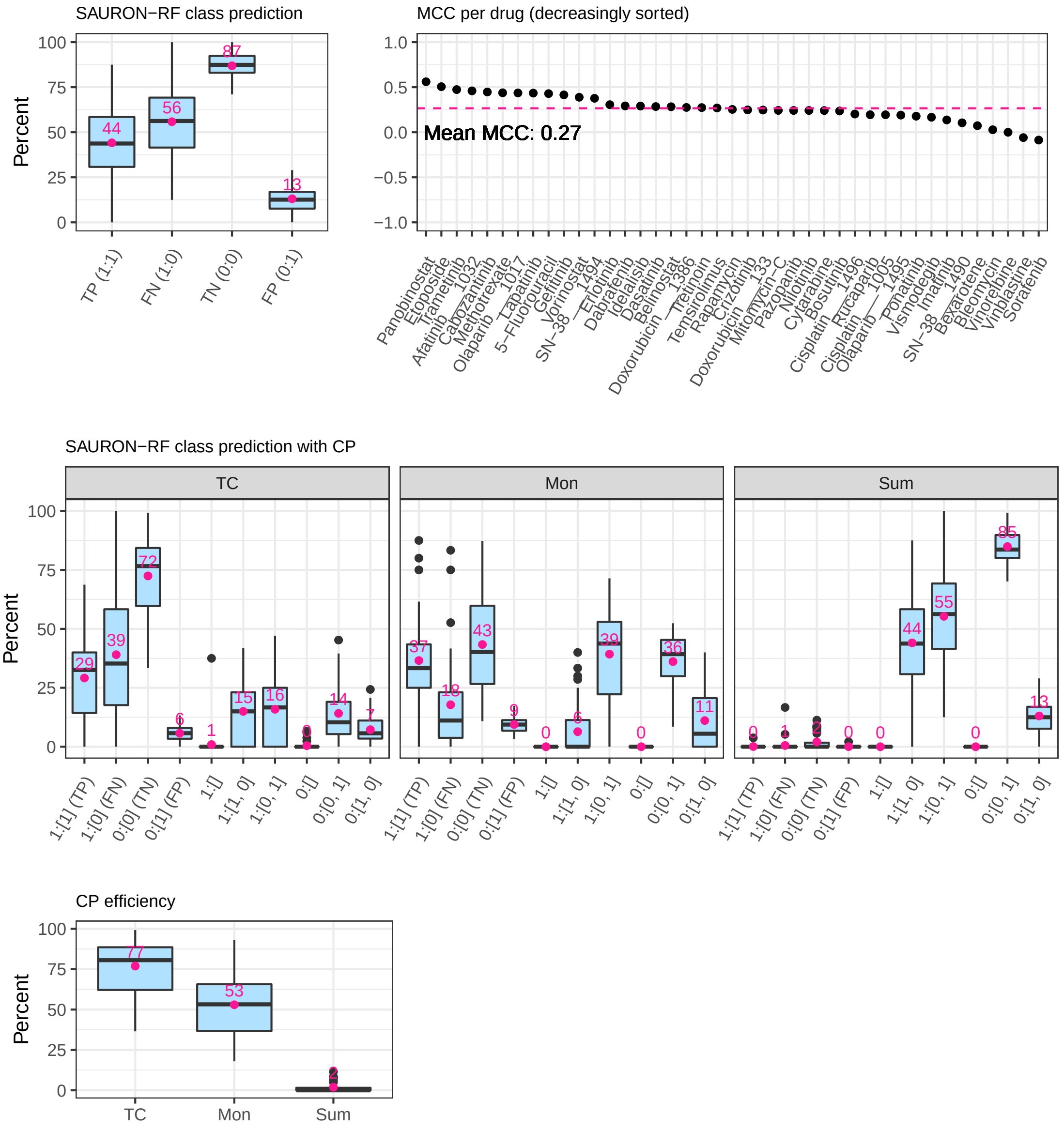


Figure E.27: Classification results for 41 GDSC1 drugs in the two-class classification setting using IC50 values. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC for each drug. The middle row shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

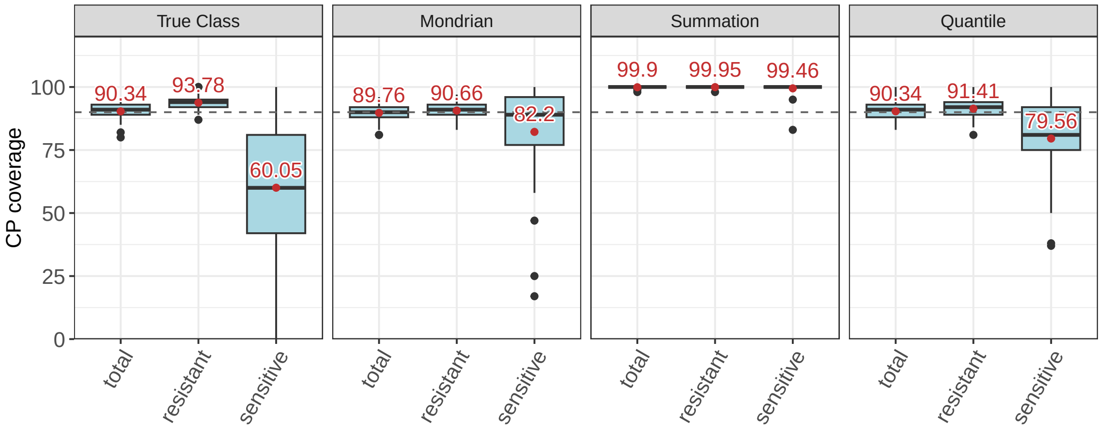


Figure E.28: Coverage evaluation for 41 GDSC1 drugs in the two-class classification setting using IC50 values. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

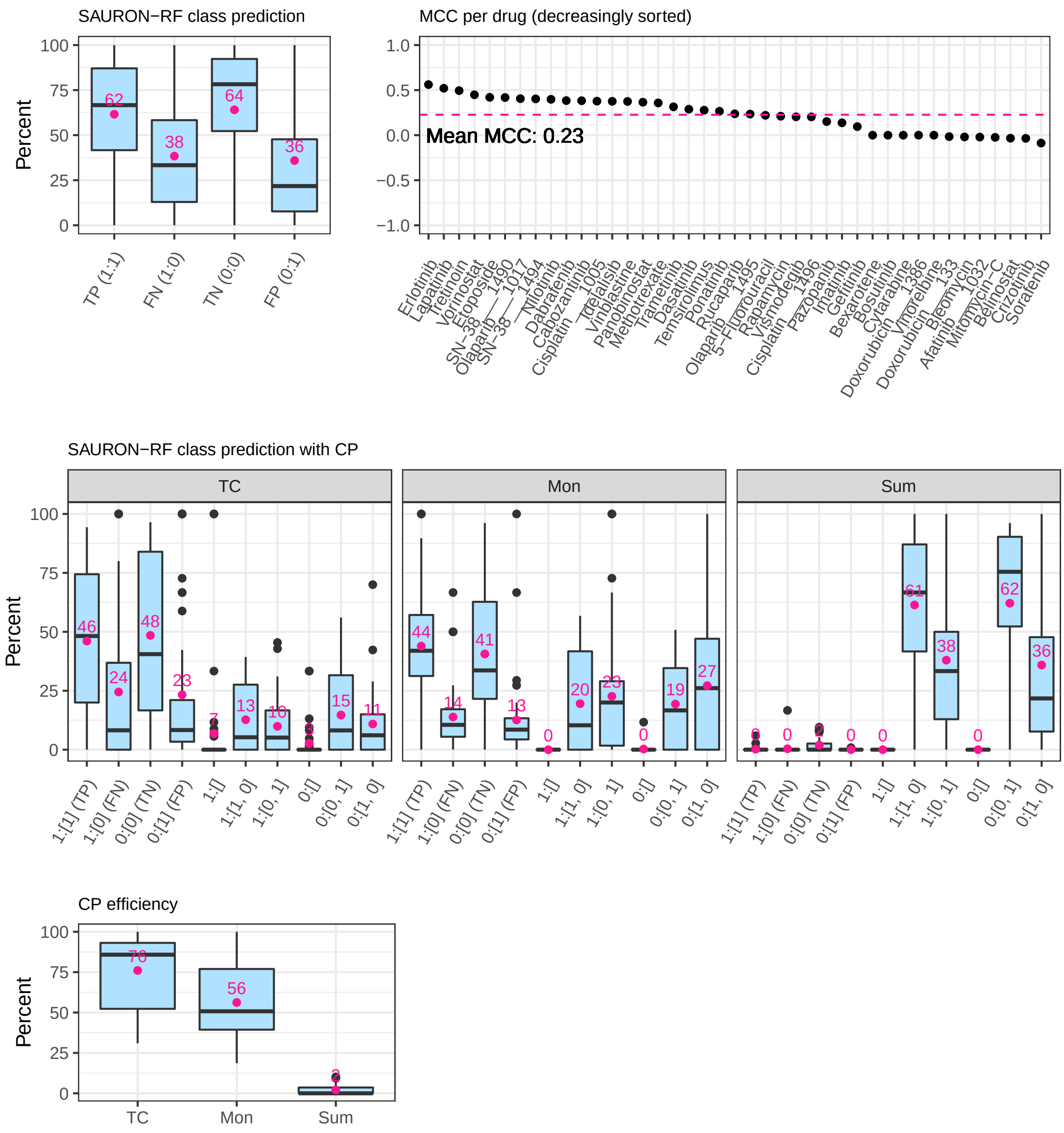


Figure E.29: Classification results for 41 GDSC1 drugs in the two-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC for each drug. The middle row shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

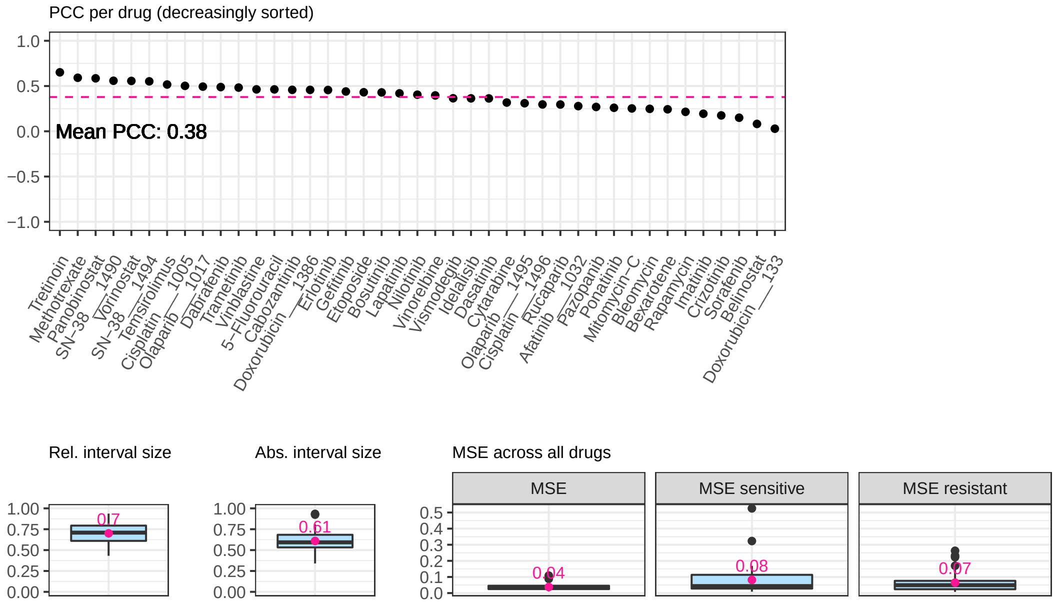


Figure E.30: Regression results for 41 GDSC1 drugs in the two-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive and resistant samples are shown.

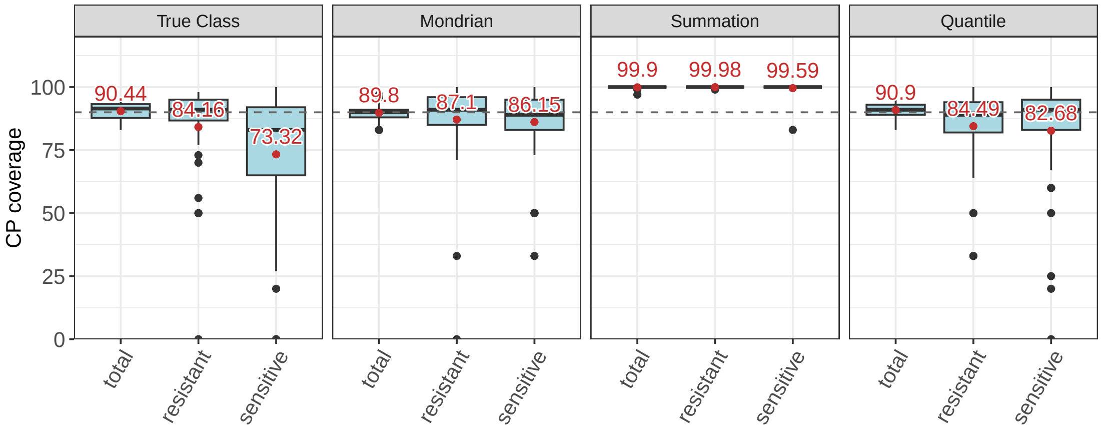


Figure E.31: Coverage evaluation for 41 GDSC1 drugs in the two-class classification setting using CMax viabilities. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

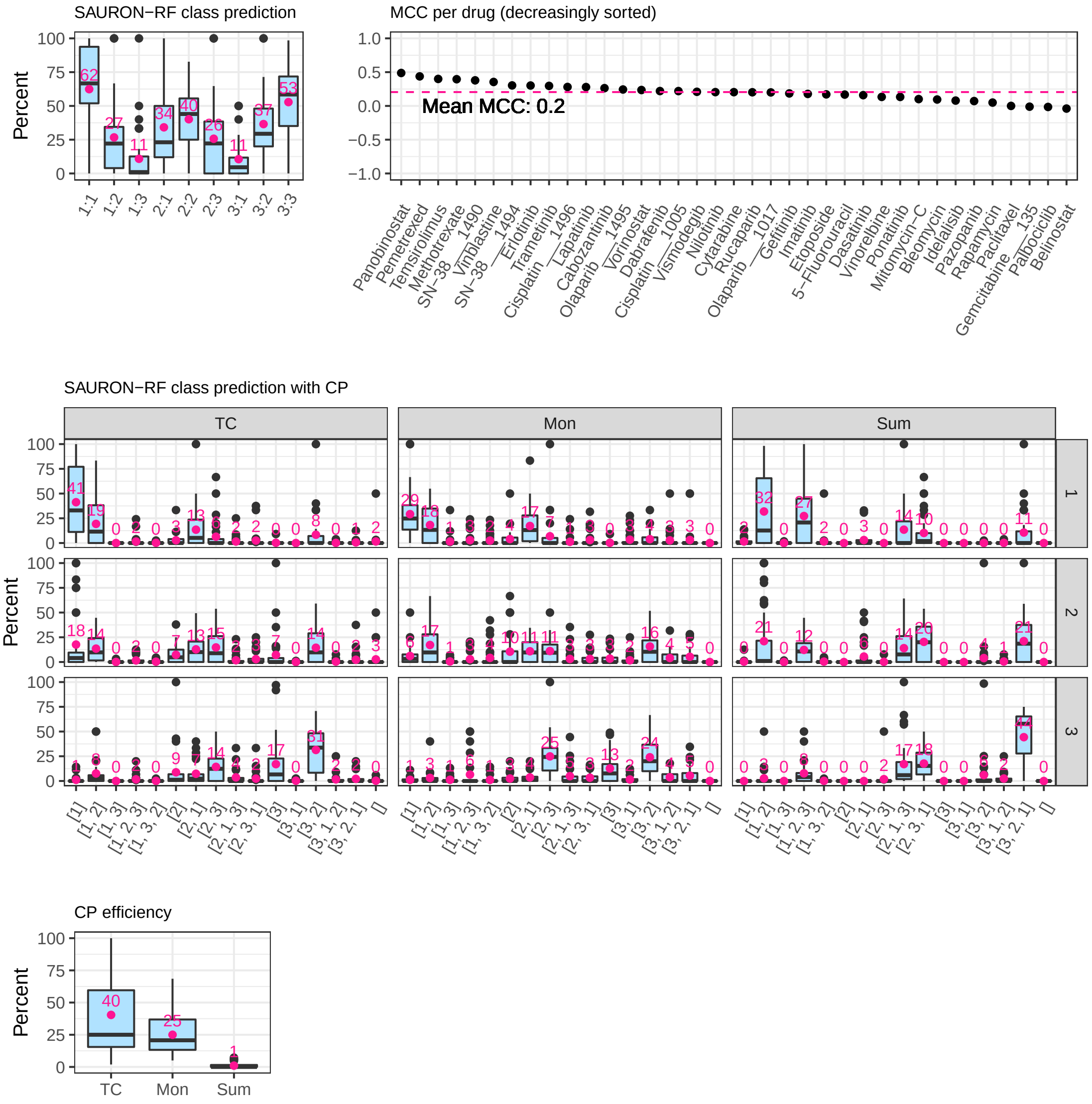


Figure E.32: Classification results for 37 GDSC1 drugs in the three-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of predictions across drugs as well as the MCC for each drug. In the left plot, each x-axis label denotes the actual and predicted class separated by a colon (1: sensitive, 2: intermediate, 3: resistant). The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). The middle row shows how predictions are affected by applying CP with three different classification scores. For two- and three-class sets, classes with higher prediction probability are listed first. In the bottom row, the CP efficiency for each score is shown.

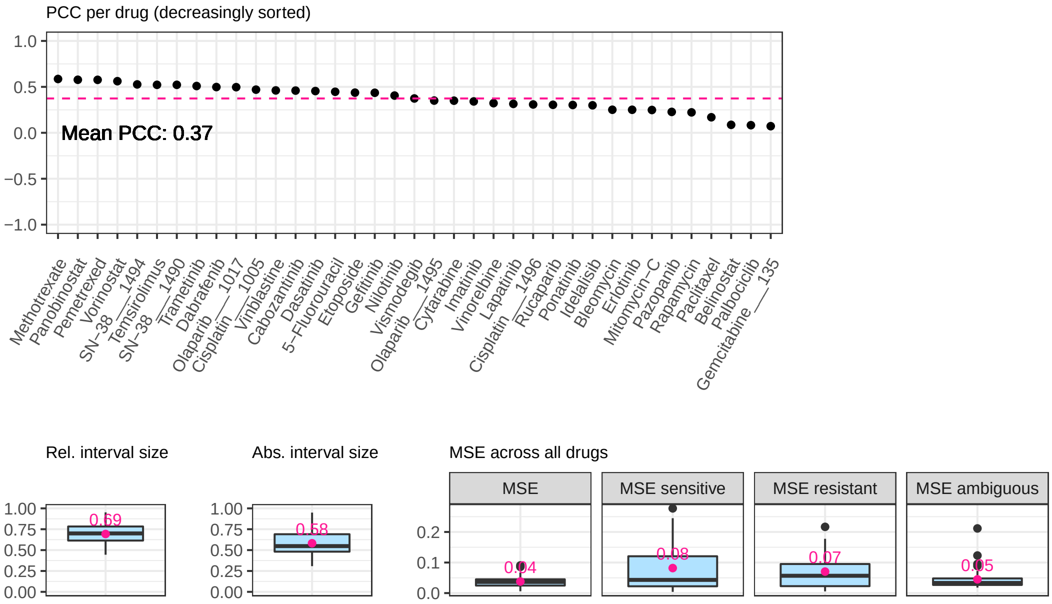


FIGURE E.33: Regression results for 37 GDSC1 drugs in the three-class classification setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive, intermediate (ambiguous), and resistant samples are shown.

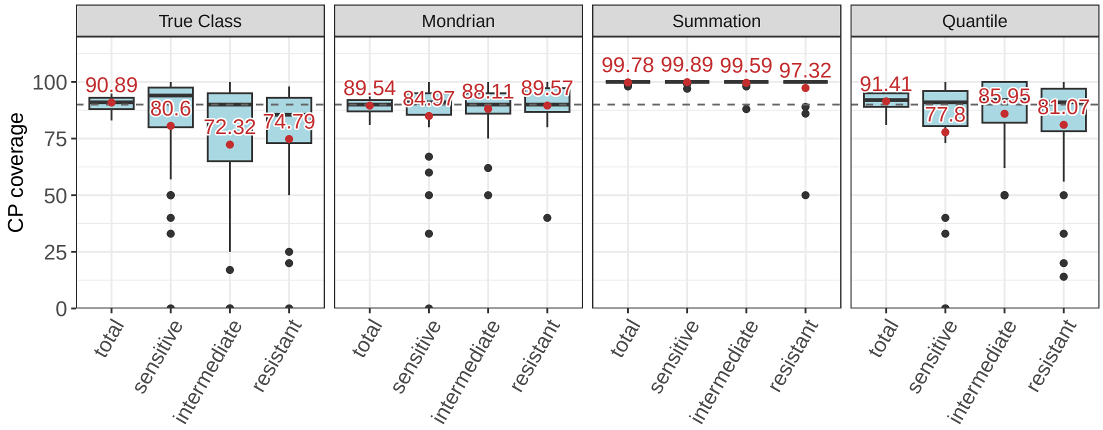


FIGURE E.34: Coverage evaluation for 37 GDSC1 drugs in the three-class classification setting using CMax viabilities. For each CP score, the total coverage as well as the coverage for the subsets of sensitive, intermediate, and resistant cell lines are shown.

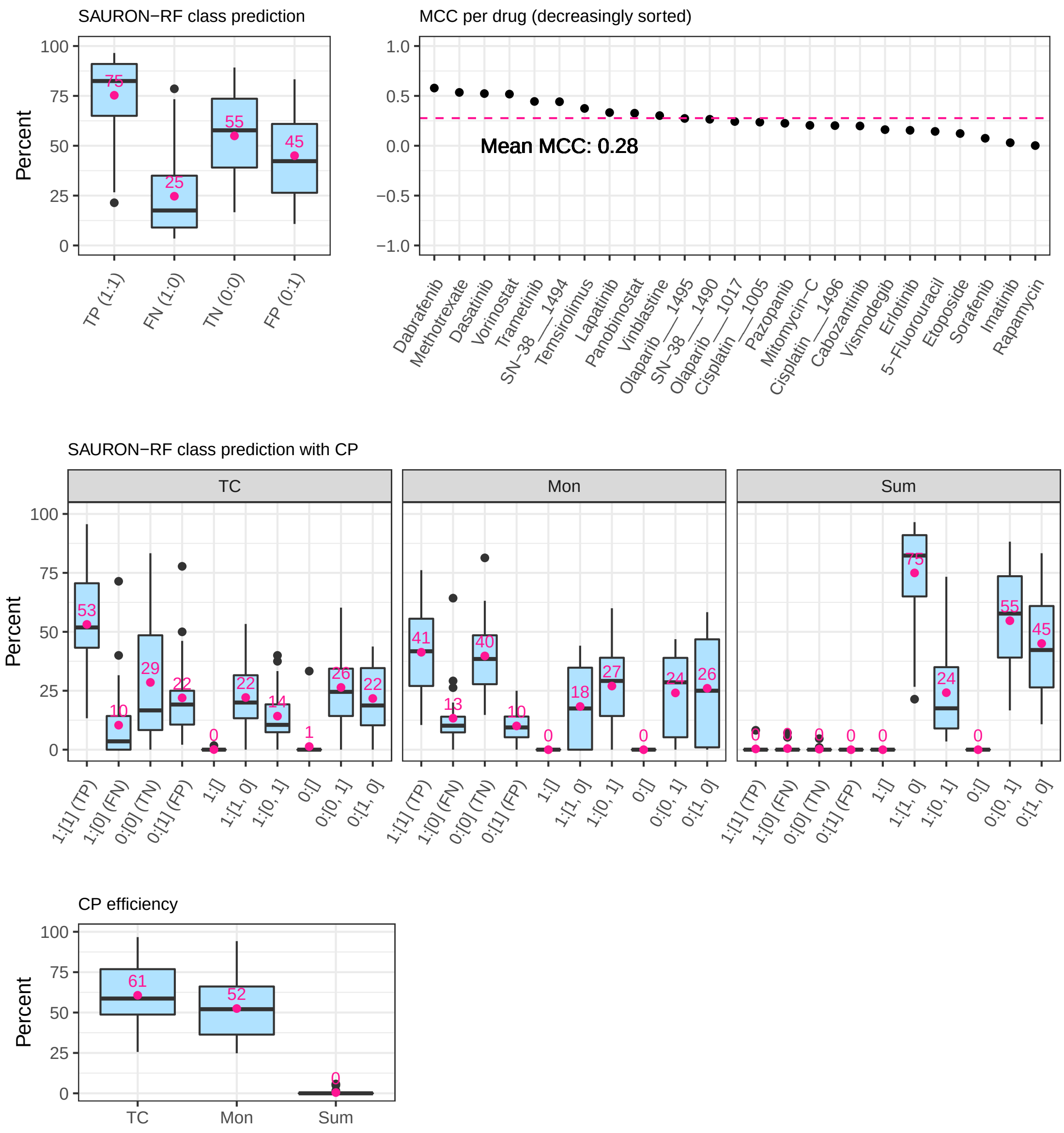


Figure E.35: Classification results for 25 GDSC1 drugs in the two-class prioritization setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF without CP and with CP. The top row depicts the performance of SAURON-RF without CP and shows the distribution of TP, FN, TN, and FP predictions across drugs as well as the MCC for each drug. The middle row shows how predictions are affected by applying CP with three different classification scores. Each x-axis label denotes the actual class and CP prediction sets separated by a colon. For two-class sets, the class with the higher prediction probability is listed first. The percentages shown on the y-axis are obtained by dividing the number of samples in each group (as defined by the x-axis) by the number of samples in the actual class (cf. Appendix E.3 for the corresponding equations). In the bottom row, the CP efficiency for each score is shown.

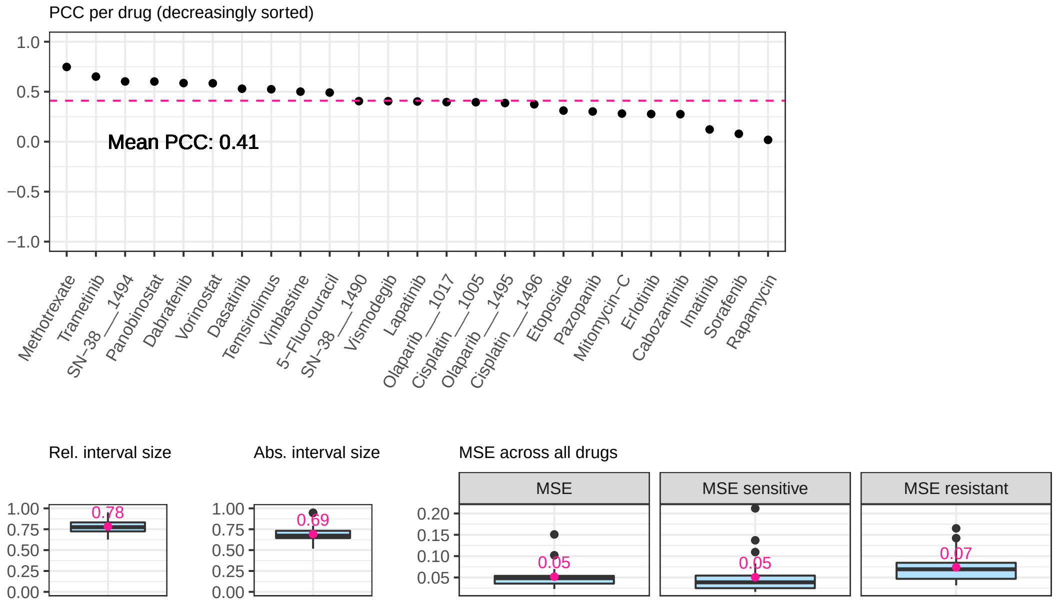


Figure E.36: Regression results for 25 GDSC1 drugs in the two-class prioritization setting using CMax viabilities. This figure depicts the test set performance of SAURON-RF with and without CP. The top row depicts the PCC between the actual and predicted values for each drug. The bottom row shows the relative and absolute interval size across drugs after applying CP using the quantile regression score. Additionally, the overall MSE as well as the MSE for the subsets of sensitive and resistant samples are shown.

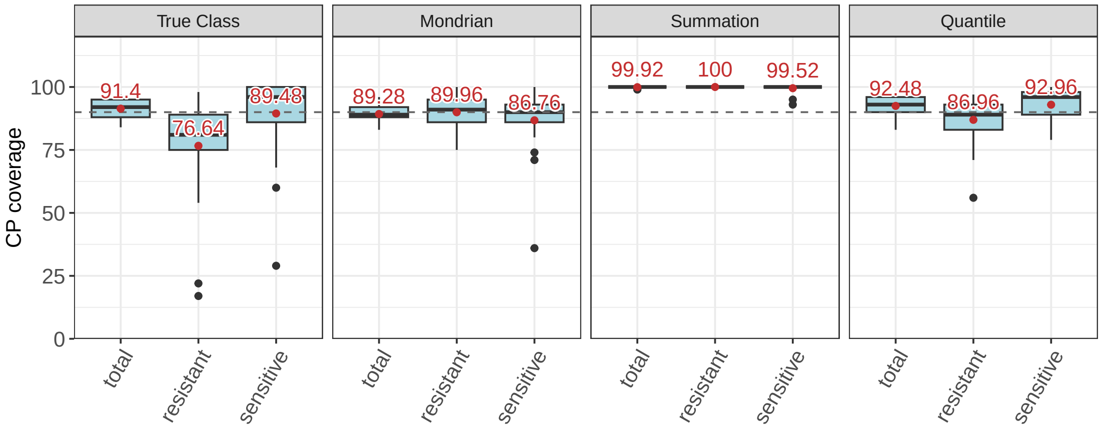


Figure E.37: Coverage evaluation for 25 GDSC1 drugs in the two-class prioritization setting using CMax viabilities. For each CP score, the total coverage and the coverage for the subsets of sensitive and resistant cell lines are shown.

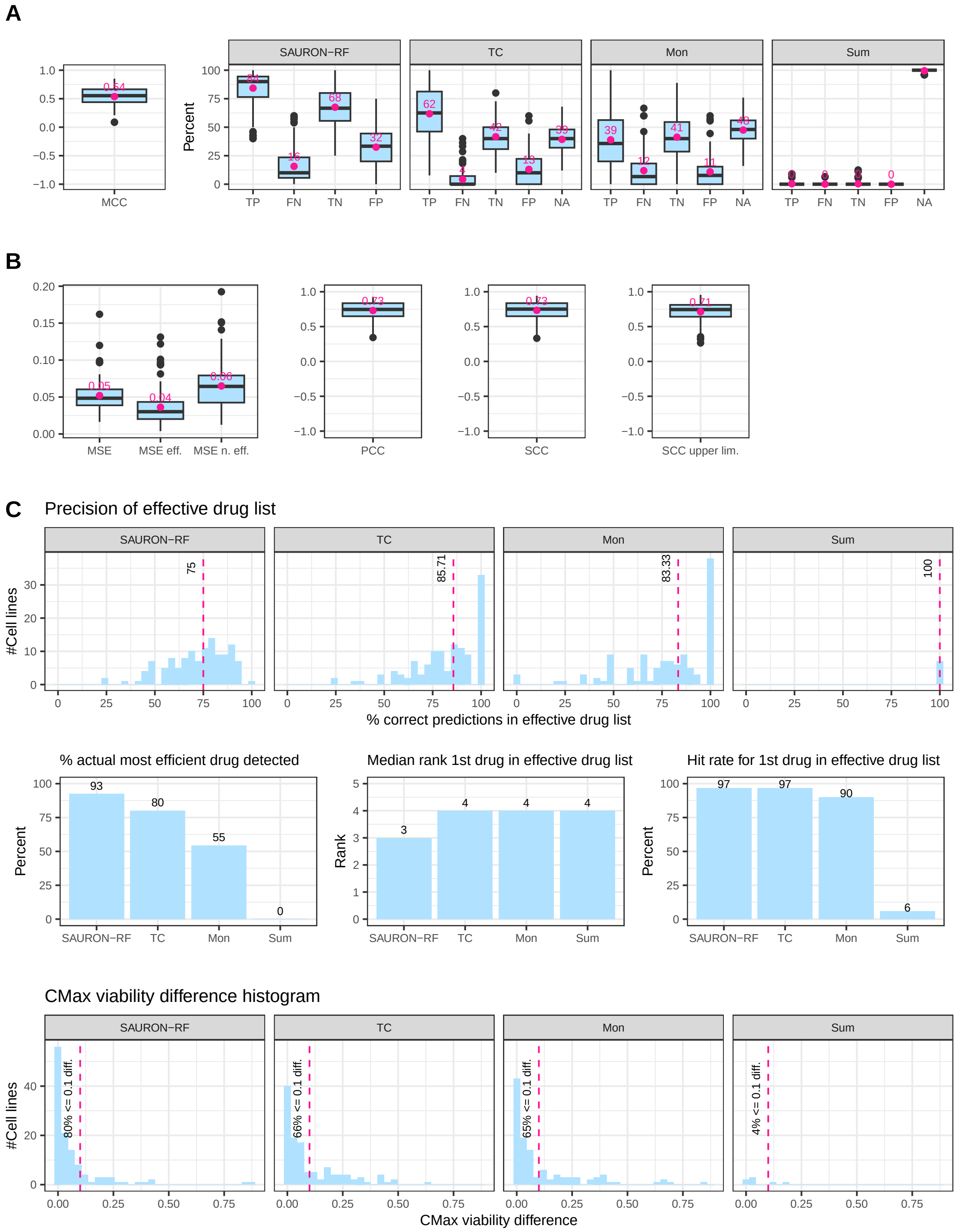


FIGURE E.38: Prioritization results for GDSC1. Subfigure A shows the classification performance of SAURON-RF. For the CP scores, single-class predictions are shown as TP, FN, TN, and FP, while all types of two-class predictions are summarized and denoted as NA. Subfigure B shows the regression performance: The average MSE is shown for all drugs and for the subsets of (in-)effective drugs. Additionally, the PCC and SCC between the actual and predicted values or the upper limit of the CP interval are shown. The top row of Subfigure C depicts the precision, i.e., the percentage of TPs in the prioritized lists generated using SAURON-RF without and with CP. The dashed red line marks the median precision. The middle row shows (1) the percentage of cases where the actual most efficient drug was contained in the prioritized list, (2) the median rank that the first drug in each predicted list has in the actual drug list, and (3) the percentage of cases for which the drug that was predicted to be most effective was actually a TP as opposed to an FP. The bottom row shows the CMax viability difference between the actual most effective drug and the drug predicted to be most effective. The dashed red line marks a difference of 0.1.

Appendix F

Combination Sensitivity Prediction

The content and text of this appendix are based on the following manuscript and its supplement, as well as the corresponding preprint, which were all written by me:

Eckhart, L., Lenhof, K., Herrmann, L., Rolli, L. M. & Lenhof H. P. (2025). *How to Predict Effective Drug Combinations - Moving beyond Synergy Scores.* Accepted for publication in iScience.
DOI (iScience pre-proof): 10.1016/j.isci.2025.112622
DOI (bioRxiv preprint): 10.1101/2024.11.22.624812

Table F.1: Investigated compounds. Table continues on next pages.

	Compound Name / CAS Number		Compound Name / CAS Number
1	Abiraterone	50	CHEMBL3103192
2	actinomycin D	51	CHIR-99021
3	ADM hydrochloride	52	chlorambucil
4	Afatinib	53	cis-Platin
5	Akt inhibitor VIII	54	Co-V
6	allopurinol	55	Crizotinib
7	alpelisib	56	cyclophosphamide
8	altretamine	57	Cylocide
9	amifostine	58	CYTARABINE HYDROCHLORIDE
10	anastrozole	59	dacarbazine
11	Antibiotic AD 32	60	Daporinad
12	Antibiotic AY 22989	61	Darinaparsin
13	Avagacestat	62	Dasatinib
14	Axitinib	63	Decitabine
15	Azacytidine, 5-	64	Deforolimus
16	AZD1208	65	dexamethasone
17	AZD1480	66	Dexrazoxane
18	AZD2014	67	Dinaciclib
19	AZD4320	68	docetaxel
20	AZD4547	69	Doramapimod
21	AZD5363	70	dorsomorphin
22	AZD5582	71	Dovitinib
23	AZD6482	72	doxorubicin
24	AZD6738	73	Elesclomol
25	AZD7762	74	Eloxatin (TN)
26	AZD8055	75	Eloxatin (TN) (Sanofi Synthelab)
27	AZD8186	76	EMBELIN
28	Belinostat	77	Emcyt (Pharmacia)
29	Bendamustine hydrochloride	78	Entinostat
30	Bexarotene	79	Enzastaurin
31	BI-78D3	80	Erlotinib
32	BI-D1870	81	Erlotinib hydrochloride
33	BI 2536	82	etoposide
34	bicalutamide	83	EXEMESTANE
35	Bleo	84	Fedratinib
36	bleomycin	85	FH535
37	BMS-536924	86	Fingolimod
38	BMS-754807	87	Fludarabine Base
39	Bortezomib	88	Foretinib
40	Bosutinib	89	Fulvestrant
41	busulfan	90	GDC-0879
42	CABAZITAXEL	91	Gefitinib
43	Cabozantinib	92	geldanamycin
44	Carboplatinum	93	gemcitabine
45	carmustine	94	GSK 650394
46	Cediranib	95	GSK429286A
47	celecoxib	96	GW 441756
48	CHEMBL17639	97	GW0742
49	CHEMBL277800	98	GW2580

Continuation of Table F.1.

	Compound Name / CAS Number
99	GW843682X
100	hydroxyurea
101	Idelalisib
102	ifosfamide
103	Imatinib
104	IMD-0354
105	IMIQUIMOD
106	IPA-3
107	Ixabepilone
108	JZL184
109	Ku-0063794
110	KU-55933
111	KU-60019
112	l-685,458
113	L-778123 free base
114	Lapatinib
115	Lenalidomide
116	Lestaurtinib
117	letrozole
118	lfm-a13
119	Linifanib
120	Linsitinib
121	lomustine
122	Masitinib
123	MEGESTROL ACETATE
124	Melphalan hydrochloride
125	metformin
126	methotrexate
127	methoxsalen
128	Midostaurin
129	MITHRAMYCIN
130	mitomycin C
131	mitotane
132	mitoxantrone
133	MK-1775
134	MK-2206
135	MK-4541
136	MK-5108
137	MLN4924
138	MRK003
139	Navelbine ditartrate (TN)
140	Navitoclax
141	Nilotinib
142	Niraparib
143	NSC-127716
144	NSC256439
145	NSC609699
146	NSC733504
147	NSC756645
148	Nutlin-3
149	Olaparib
150	Onalespib
151	OSI-027
152	OSI-930
153	OSU-03012
154	paclitaxel
155	Palbociclib
156	Panobinostat
157	parthenolide
158	Pazopanib
159	Pazopanib hydrochloride
160	Pemetrexed
161	Perifosine
162	PF-04217903
163	PF-562271
164	PHA-793887
165	PI-103
166	PIK-93
167	Pioglitazone
168	Piperlongumine
169	pipobroman
170	PLX-4720
171	Pralatrexate
172	Procarbazine hydrochloride
173	QS11
174	Quinacrine hydrochloride
175	Quizartinib
176	RAF265
177	raloxifene
178	Retinoic acid
179	Romidepsin
180	Ruxolitinib
181	Sapitinib
182	Saracatinib
183	Selumetinib
184	Serdemetan
185	Silmitasertib
186	SNS-032
187	SNX-2112
188	Sorafenib
189	Sunitinib
190	T0901317
191	Tamoxan
192	Tamoxifen citrate
193	Tanespimycin
194	temozolomide
195	Temsirolimus
196	teniposide

Continuation of Table F.1.

	Compound Name / CAS Number
197	TGX-221
198	thalidomide
199	thapsigargin
200	thiotepa
201	Tipifarnib
202	Tivozanib
203	topotecan
205	Tozasertib
206	TPCA-1
207	Trametinib
209	Trisenox
210	Tubastatin A
211	TW-37
212	UNC0638
213	Uramustine
214	US9505780, JQ-1
215	Vandetanib
216	Veliparib
217	Vemurafenib
218	Vepesid J
219	vinblastine
220	Vinblastine sulfate
221	vincristine
222	Vincristine sulfate
223	vinorelbine
224	Vismodegib
225	Vorinostat
226	VX-702
227	XL147
228	XL765
229	YK 4-279
230	Zanosar
231	Zoledronic acid
232	ZSTK474
233	(-)-Rapamycin
234	001, RAD
235	1032350-13-2
236	122111-05-1
237	1260907-17-2
238	158798-73-3
239	218137-86-1
240	219580-11-7
241	23541-50-6
242	284028-89-3
243	303727-31-3

	Compound Name / CAS Number
244	315183-21-2
245	391210-10-9
246	49843-98-3
247	5-Aminolevulinic acid hydrochloride
248	5-azacytidine
249	5-Fluoro-2'-deoxyuridine
250	5-Fluorouracil
251	547757-23-3
252	55-86-7
253	6-Mercaptopurine
254	6-Thioguanine
255	7-Ethyl-10-hydroxycamptothecin
256	717906-29-1
257	761439-42-3
258	7803-88-5
259	781661-94-7
260	803712-79-0
261	841290-80-0
262	844499-71-4
263	891494-63-6
264	915019-65-7
265	957054-30-7

Table F.2: Hyperparameters of the investigated ML algorithms. This table denotes the tuned hyperparameters for each ML algorithm. For hyperparameters not stated explicitly, the default parameters as provided the respective Python package were employed. Explicitly tuned hyperparameters are marked in bold. For the PhysChem setting (i.e., the setting with the largest data matrix), we were unable to train neural networks with the ELU activation or learning rates of 0.1 due to insufficient memory for resource allocation even when decreasing the batch size.

Model	Parameter	Value(s)
Elastic net	**alpha**	0.01, 0.1, 1, 10, 100
	l1_ratio	0, 0.25, 0.5, 0.75, 1
Random forest	**max_depth**	100, 1000000
	max_features	25, 50, 100, 250
	min_samples_leaf	2, 20, 100, 1000
	n_estimators	500
Neural network	loss	mean_squared_error
	activation	tanh, ELU (none in last layer)
	optimizer	Adam
	learning_rate	0.0001, 0.001, 0.1
	hidden_layers	1,2,3,4,5
	size of hidden layers	equally spaced btw. in-/output size
	dropout	0.1, 0.3
	batch_size	256
	bias_initializer	0.01
	kernel_initializer	glorot_uniform for tanh, he_normal for ELU activation
	kernel_regularizer	l2
	epochs	300
	validation_split	0.2
	early stopping	yes
	patience	15
	restore_best_weights	True

Table F.3: This table denotes the tuned hyperparameters for each ML algorithm and setting. For hyperparameters not stated explicitly, the values denoted in Appendix Table F.2 were employed. Otherwise, we used the default parameters as provided the respective Python packages.

Model	Setting	Parameters

Random Forest	OneHot	max_features=250; max_depth=1000000;
		min_samples_leaf=2

Random Forest	OneHotTar	max_features=250; max_depth=1000000;
		min_samples_leaf=2

Random Forest	MACCS	max_features=250; max_depth=100;
		min_samples_leaf=2

Random Forest	PhysChem	max_features=100; max_depth=100;
		min_samples_leaf=2

Neural Network	OneHot	activation=elu; learning_rate=0.0001;
		num_hidden_layers=5; dropout=0.3;

Neural Network	OneHotTar	activation=elu; learning_rate=0.0001;
		num_hidden_layers=4; dropout=0.3;

Neural Network	MACCS	activation=elu; learning_rate=0.0001;
		num_hidden_layers=5; dropout=0.3;

Neural Network	PhysChem	activation=tanh; learning_rate=0.0001;
		num_hidden_layers=4; dropout=0.1;

Elastic Net	OneHot	alpha=0.01; l1_ratio=1
Elastic Net	OneHotTar	alpha=0.01; l1_ratio=1
Elastic Net	MACCS	alpha=0.01; l1_ratio=1
Elastic Net	PhysChem	alpha=0.01; l1_ratio=1

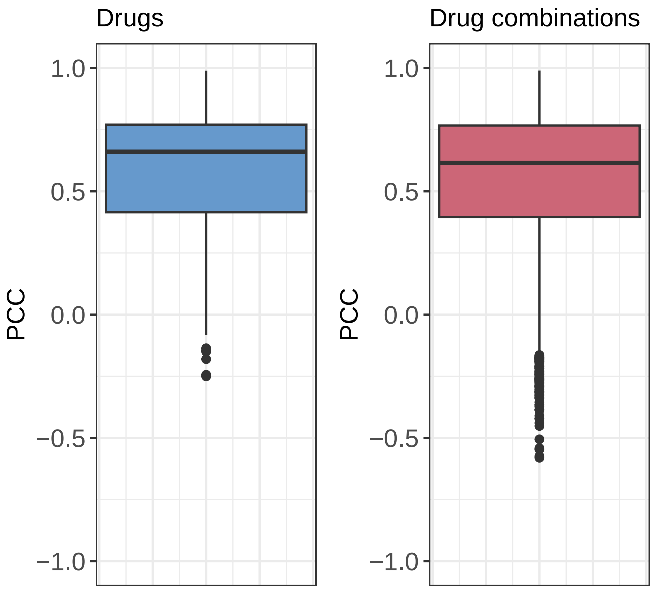


Figure F.1: Average Pearson correlation between actual and predicted relative inhibitions per drug (blue) and per drug combination (red) using the MACCS random forest.

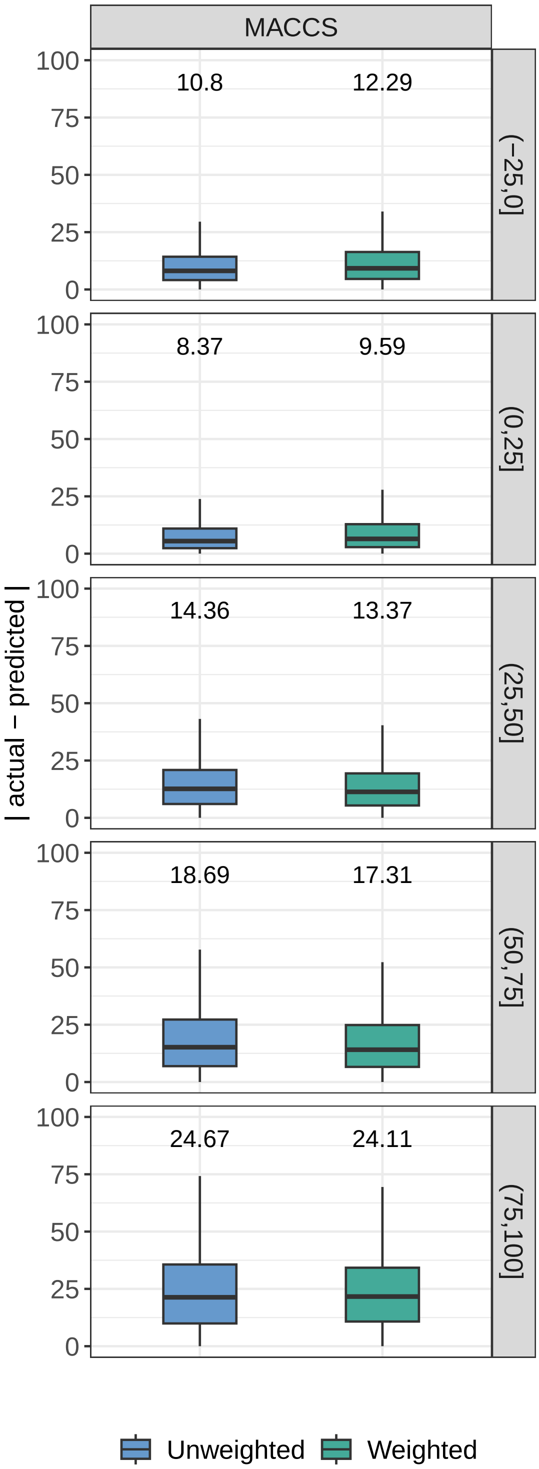


FIGURE F.2: Impact of sample weights on model predictions. This figure compares the prediction performance (in terms of the absolute difference between actual and predicted values) for the random forest MACCS model with (green) and without (blue) sample weights. Each row shows the performance for a different interval of actual relative inhibitions. On top of each boxplot, the MAE is shown. The weight of each sample was determined based on its interval $i \in I = \{(-\infty, 0], (0, 25], (25, 50], (50, 75], (75, \infty)\}$ as $((max_{j \in I}|j|)/|i|)^2$, where $|i|$ denotes the number of training samples in interval i.

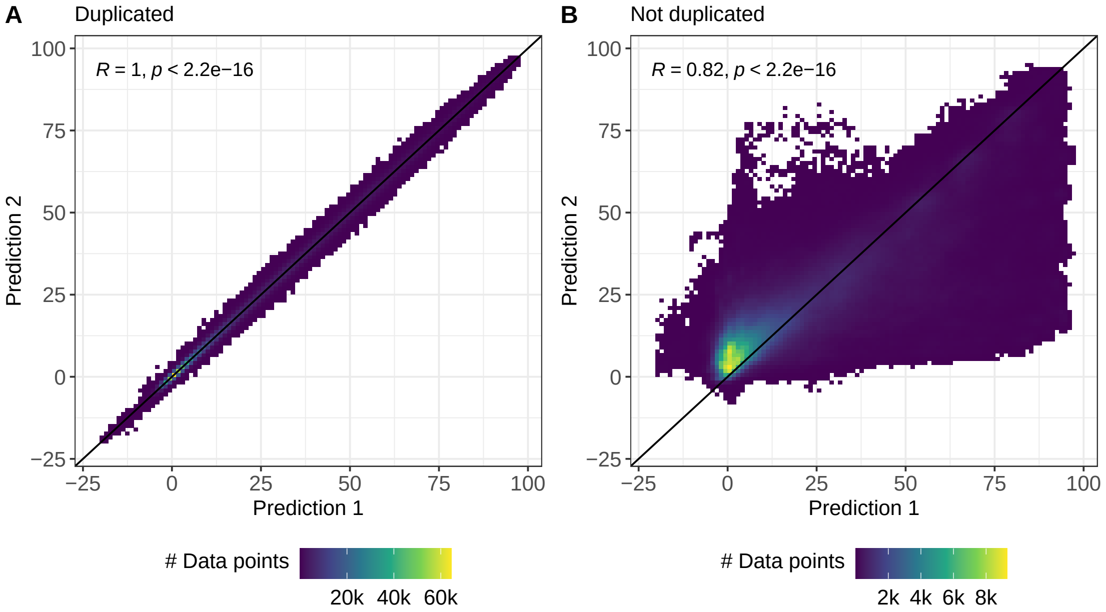


FIGURE F.3: Correlation of duplicated entries from the test data. This figure shows the correlation between the predictions for duplicated entries obtained from the random forest PhysChem model. Duplicated entries refer to the same drug-drug-cell combination and the same treatment concentrations but can be represented by two different model inputs through swapping the features of the respective drugs (cf. Figure 8.1). Subfigure A shows the test predictions when including duplicated entries in the training data, while Subfigure B shows the predictions when training only on non-duplicated entries. In both figures, the black diagonal line represents the identity and R denotes the Pearson correlation between the predictions.

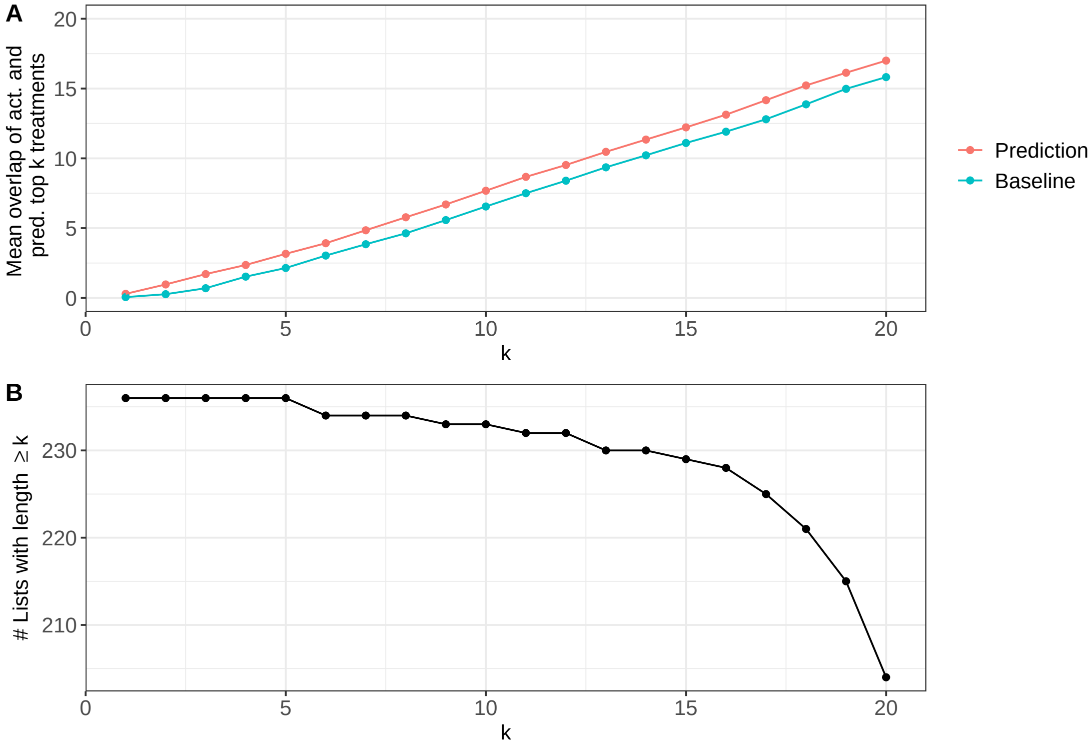


Figure F.4: Overlap of k actual and predicted best treatments for monotherapies. Subfigure A shows the average intersection size between the k actual best treatments and the k predicted best treatments for each cell line. Subfigure B shows the number of cell lines based on which the average for each k was computed.

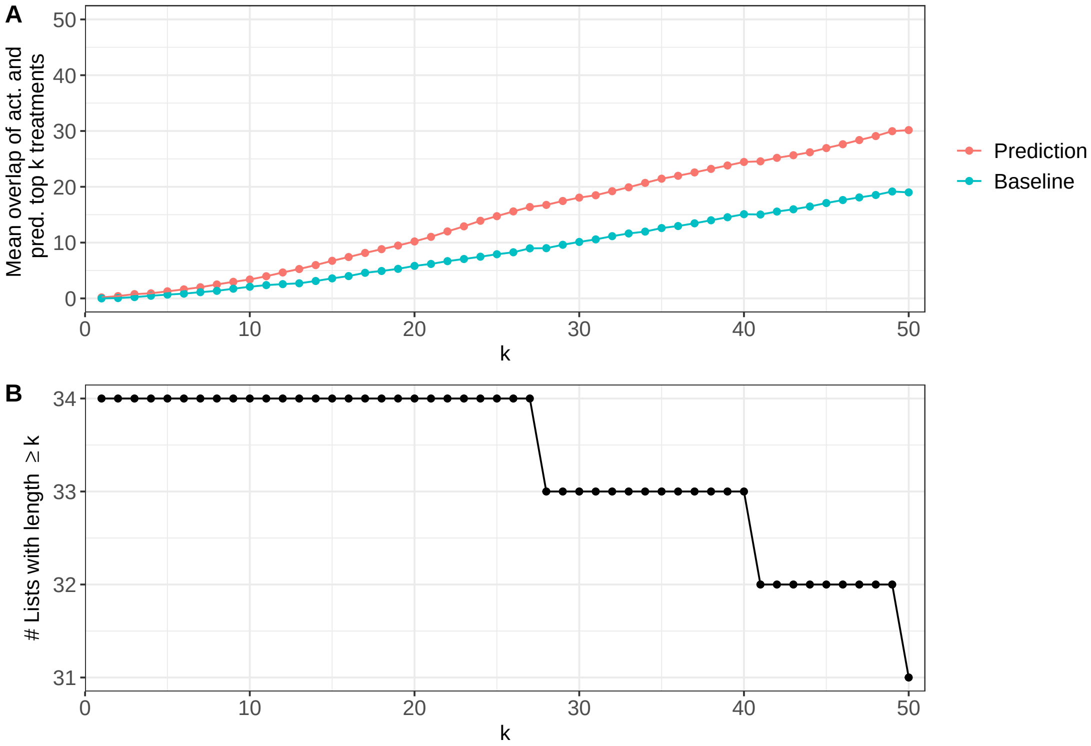


Figure F.5: Overlap of k actual and predicted best treatments for the combination of both mono- and combination therapies. Subfigure A shows the average intersection size between the k actual best treatments and the k predicted best treatments for each cell line. Subfigure B shows the number of cell lines based on which the average for each k was computed. Note that the number of test cell lines with available combination data is smaller than the number of cell lines with available monotherapy data and we show the results only for cell lines where both were available.

F.1 Reconstructing Sensitivity Measures: Additional Analyses

Hypothesis 1: The CMax viability is difficult to predict since concentrations exceeding the CMax concentration were not screened for 14 of 77 drugs, corresponding to 30% of the drug-cell line combinations in the test set. Thus, the curve-fitting for these entries might not accurately model the CMax viability.

Evaluation 1: We evaluated the PCC only on those drug-cell line combinations for which a concentration greater than CMax was screened. This increased the average PCC slightly from 0.1 to 0.12.

Hypothesis 2: Larger prediction errors for high inhibitions (i.e., low viabilities) make the curve-fitting unreliable in areas of high inhibition (cf. Figure 8.4), which affects the derived response measures.

Evaluation 2: The IC50 value is designed to measure the drug response at a relative inhibition of 50%. To assess the performance at smaller inhibitions (i.e., higher viabilities), we reconstructed IC75 and IC90 values from the fitted curves. The IC75 (IC90) measures the drug concentration where a relative inhibition of $100\% - 75\% = 25\%$ ($100\% - 90\% = 10\%$) is reached. The average per-drug PCCs for the IC75 and IC90 reconstruction are 0.11 and 0.1, respectively, thus, there is no (strong) improvement compared to the IC50 predictions (cf. Appendix Figure F.6).

Hypothesis 3: The two-step process of first reconstructing a curve and then deriving the CMax viability from the curve is inferior to directly predicting the CMax viability with our models.

Evaluation 3: Instead of deriving the CMax viability from the estimated dose-response curves, we used our model directly to predict the relative inhibition at the CMax concentration and converted this prediction into a relative viability. However, this did also not improve correlations (PCC 0.04, cf. Appendix Figure F.7). Instead, the curve fitting seems to enhance predictions slightly, which is in line with the findings by Rahman and Pal [387].

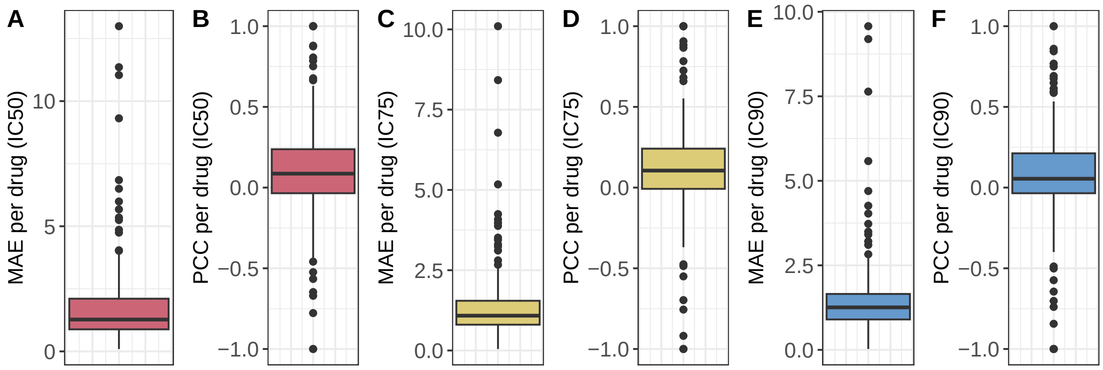


FIGURE F.6: Reconstruction of IC50, IC75, and IC90 values from model predictions. Subfigures A and B (red) show the distribution of MAE and PCC per drug for the reconstruction of IC50 values using the test set monotherapy data. Subfigures C and D (yellow) show the analogous results for IC75 values. Subfigures E and F (blue) show the analogous results for IC90 values.

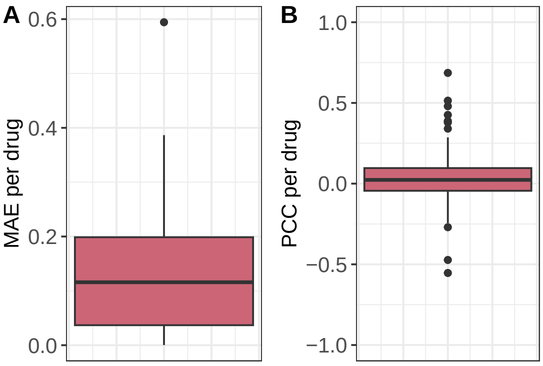


FIGURE F.7: Direct prediction of CMax viabilities. This figure shows the distribution of MAE (A) and PCC (B) per drug for the direct prediction of CMax viabilities using the MACCS random forest and the cell line-drug combinations from the test set monotherapy data.

Appendix G

Bladder Cancer Classification

The content of this appendix is based on the following publication and its supplement:

Eckhart, L.*, Rau, S.*, Eckstein, M., Stahl, P. R., Ayoubian, H., Heinzelbecker, J., ... & Junker, K. (2025) **Machine learning accurately predicts muscle-invasion of bladder cancer based on three miRNAs**. Journal of Cellular and Molecular Medicine, 29(3), e70361
DOI: 10.1111/jcmm.70361
(* authors contributed equally)

G.1 Details on the Quantification of miRNA Expression

Tumor areas were dissected from five to ten formalin-fixed, paraffin-embedded (FFPE) sections with a thickness between 7 and 10μm. The miRNAs were isolated using the Quiagen miRNeasy FFPE kit. Quantitative real-time polymerase chain reaction (qRT-PCR) was performed as follows: First, reverse transcription was performed using TaqMan® MicroRNA Reverse Transcription Kits. Next, the PCR was performed in triplicates using TaqMan® miRNA primers and TaqMan® Fast Advanced Master Mix and the Roche Diagnostics LightCycler® 480.

Table G.1: Overview of ML algorithms, including the used R packages and tuned hyperparameters. Unless stated otherwise, we employed the default parameters of each algorithm in their respective package.

Model	Parameter	Value(s)
Boosting trees (AdaBoost) (ada, v. 2.0.5 [438])	iter	1 - 5
	maxdepth	5 - 25 in steps of 5
K-nearest neighbors (class, v. 7.3.19 [508])	k	1 - 25 in steps of 2
Random forest (randomForest, v. 4.7.1.1 [361])	mtry	1 - 4
Support vector machine (kernlab v.0.9-31 [439])	kernel	linear, polynomial, radial
	C	0.001, 0.01, 0.1, 1, 10, 100
	degree (polyn. kernel)	2, 3, 4
	scale (polyn. kernel)	0.1, 0.5, 1, 2
	sigma (radial kernel)	0.01, 0.1, 0.5, 1, 5, 10
Neural network (nnet, v. 7.3.16 [508])	size	1 - 10
	decay	0, 1e-7 - 1e-2, 0.1, 0.5

Table G.2: Hyperparameters of best-performing models. Unless stated otherwise, we employed the default parameters of each algorithm in their respective package (cf. Appendix Table G.1).

Model	Parameter	Value(s)
K-nearest neighbors	k	19
Support vector machine	kernel	polynomial
	C	1
	degree	2
	scale	0.1
Neural network	size	4
	decay	0.01

Table G.3: P-values for Mann-Whitney U tests of differential expression between the MIBC and pTa lg groups for each miRNA and cohort.

Cohort	Comparison	138-4p	146b-5p	155-5p	200a-3p
1	pTa lg vs. TUR-MIBC	0.373	<0.0001	0.0002	0.1216
1	pTa lg vs. CYS-MIBC	0.0003	<0.0001	<0.0001	0.0006
1	pTa lg vs. all MIBC	0.004	<0.0001	<0.0001	0.002
2	pTa lg vs. TUR-MIBC	0.2733	<0.0001	<0.0001	0.4309
2	pTa lg vs. CYS-MIBC	<0.0001	<0.0001	<0.0001	<0.0001
2	pTa lg vs. all MIBC	<0.0001	<0.0001	<0.0001	<0.0001

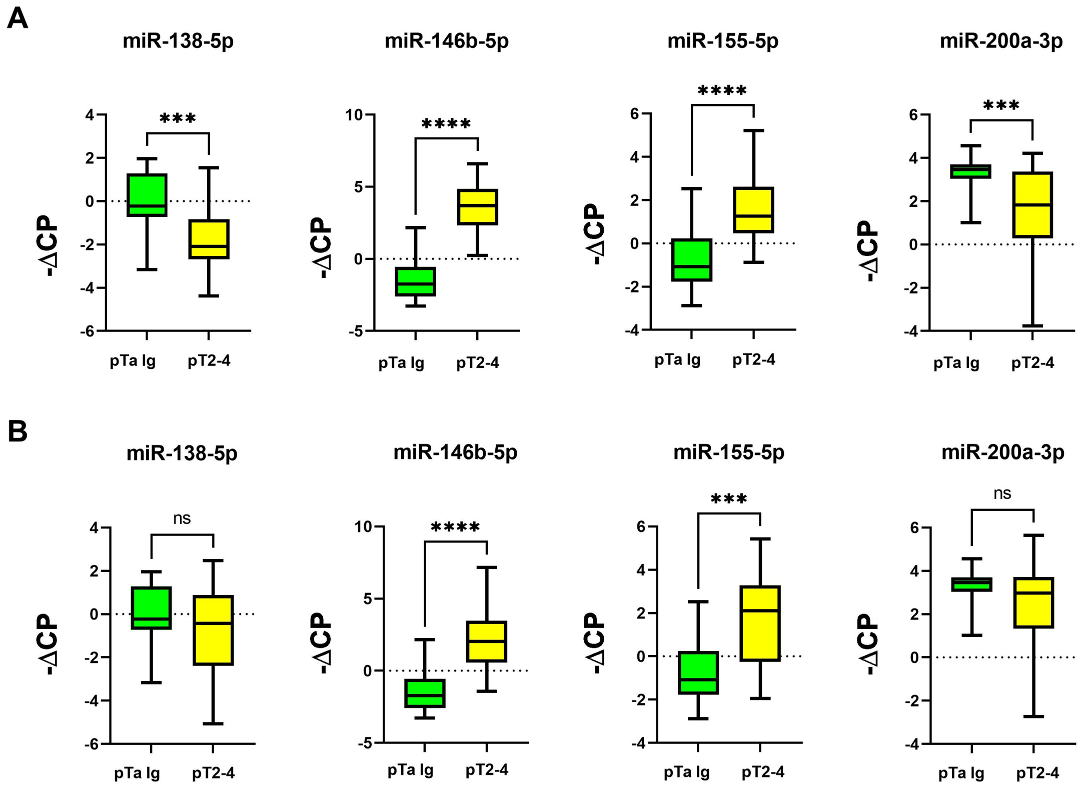


Figure G.1: Expression of miRNAs in TUR-MIBC (A) and CYS-MIBC (B) vs. pTa lg cancers in Cohort 1. For each miRNA, its expression is shown as $-\Delta$CP values for the MIBC (denoted as pT2-4) and pTa lg samples. Significant expression differences between the two groups were identified using Mann-Whitney U tests. The corresponding p-values (p) are annotated as $***$ for $p \leq 0.001$, as $****$ for $p \leq 0.0001$, and as *ns* for $p > 0.05$. They can be found in Appendix Table G.3.

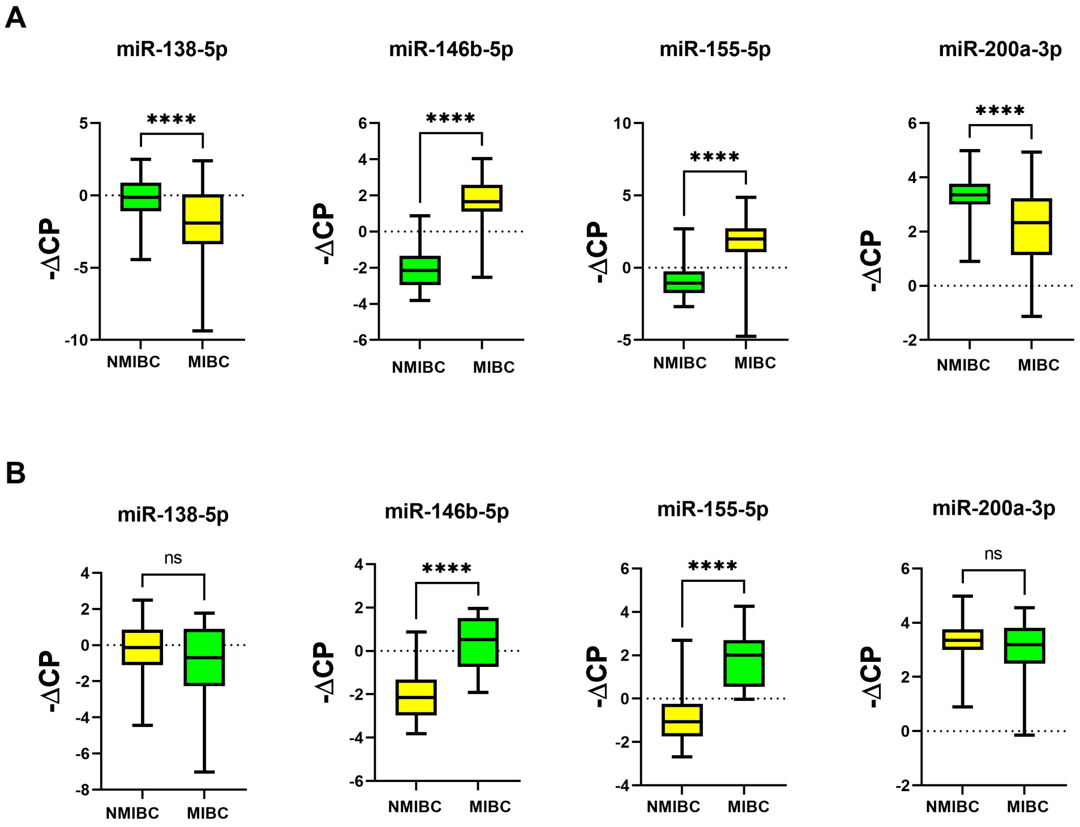


FIGURE G.2: Expression of miRNAs in TUR-MIBC (A) and CYS-MIBC (B) vs. pTa lg cancers in Cohort 2. For each miRNA, its expression is shown as −ΔCP values for the MIBC (denoted as pT2-4) and pTa lg samples. Significant expression differences between the two groups were identified using Mann-Whitney U tests. The corresponding p-values (p) are annotated as $****$ for p $\leq$ 0.0001 and as *ns* for p $>$ 0.05. They can be found in Appendix Table G.3.

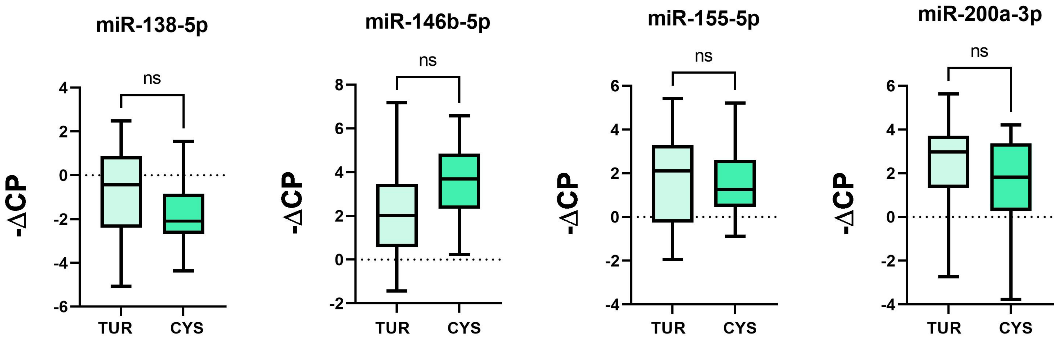


FIGURE G.3: Expression of miRNAs in TUR-MIBC vs. CYS-MIBC cancers in Cohort 1. For each miRNA, its expression is shown as −ΔCP values for the TUR-MIBC and CYS-MIBC samples. Using Mann-Whitney U tests, no significant expression differences between the two groups were identified, which is denoted as *ns* in the figure.

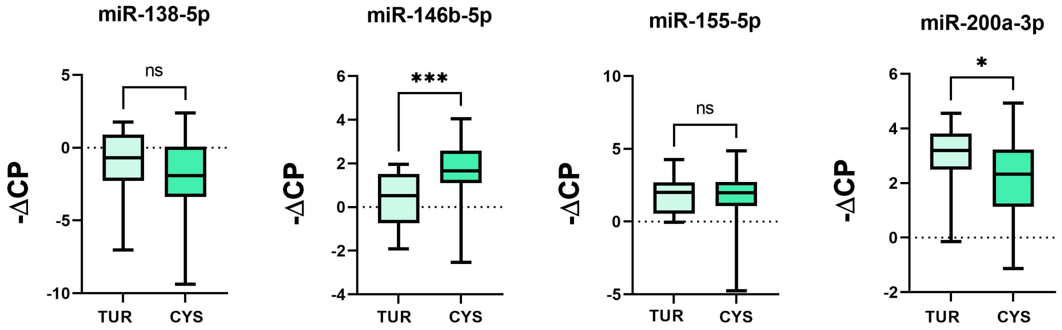


Figure G.4: Expression of miRNAs in TUR-MIBC vs. CYS-MIBC cancers in Cohort 2. For each miRNA, its expression is shown as $-\Delta$CP values for the TUR-MIBC and CYS-MIBC samples. Significant expression differences between the two groups were identified using Mann-Whitney U tests. The corresponding p-values (p) are annotated as $*$ for $p \leq 0.05$, as $***$ for $p \leq 0.001$, and as *ns* for $p > 0.05$.

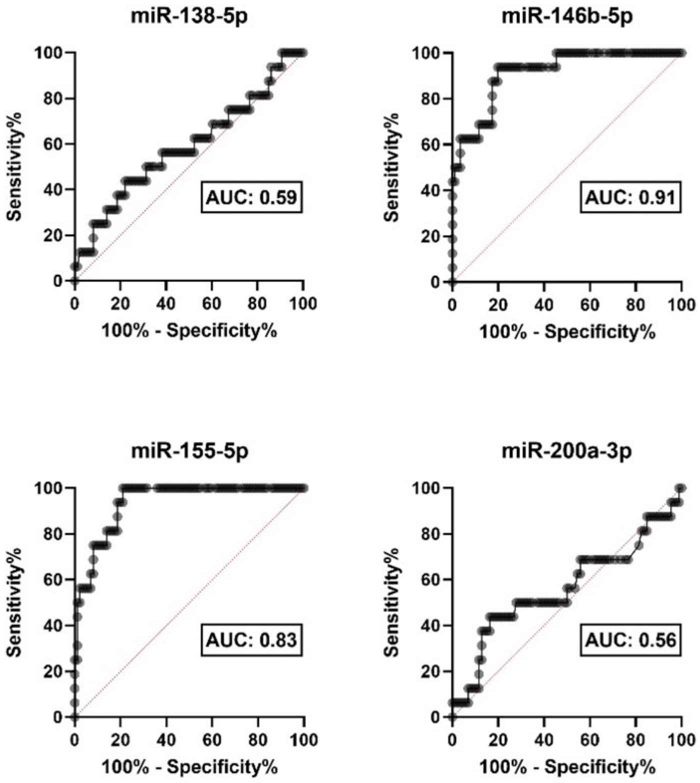


Figure G.5: ROC curves that distinguish TUR-MIBC from pTa lg in Cohort 1.

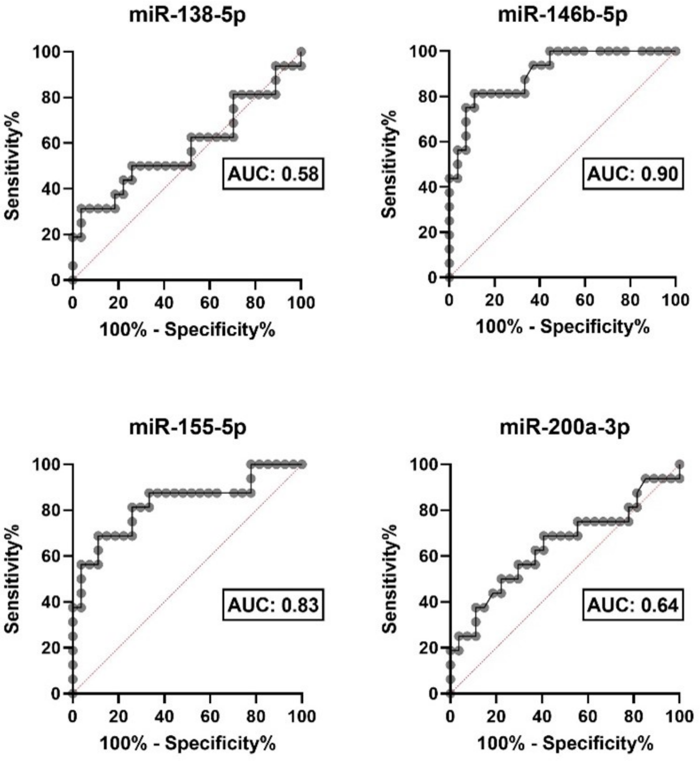


Figure G.6: ROC curves that distinguish TUR-MIBC from pTa lg in Cohort 2.

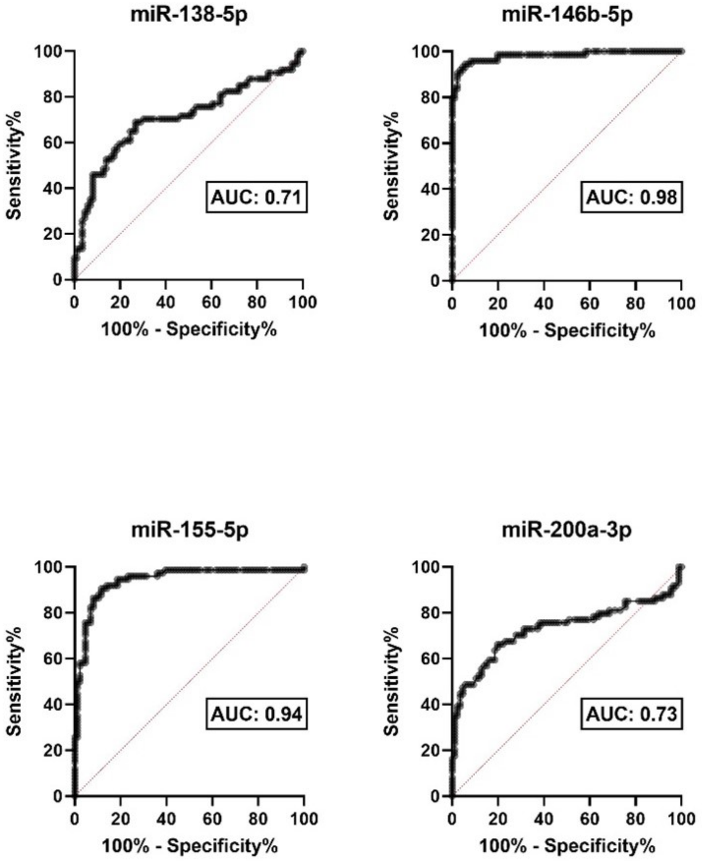


Figure G.7: ROC curves that distinguish CYS-MIBC from pTa lg in Cohort 1.

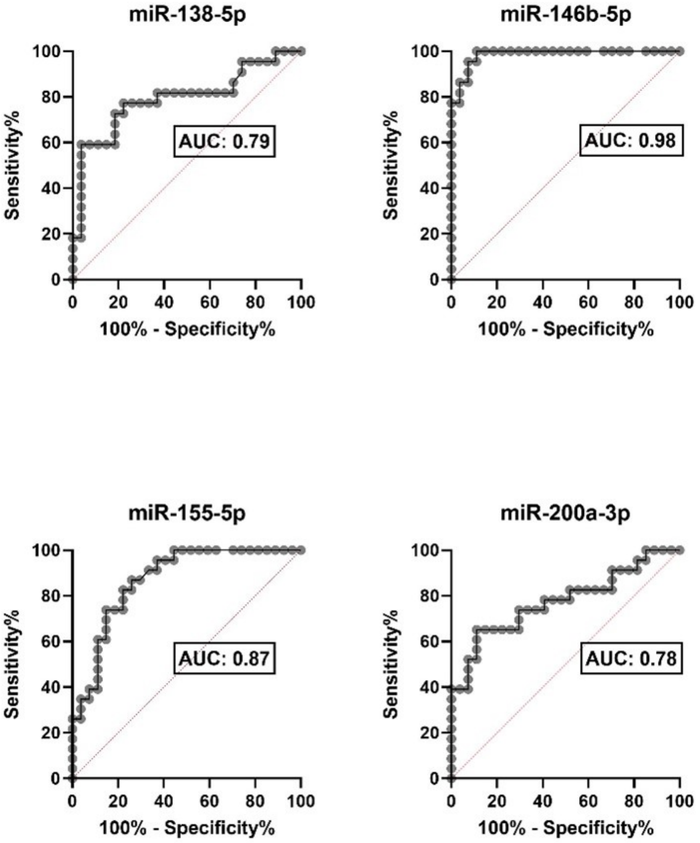


Figure G.8: ROC curves that distinguish CYS-MIBC from pTa lg in Cohort 2.

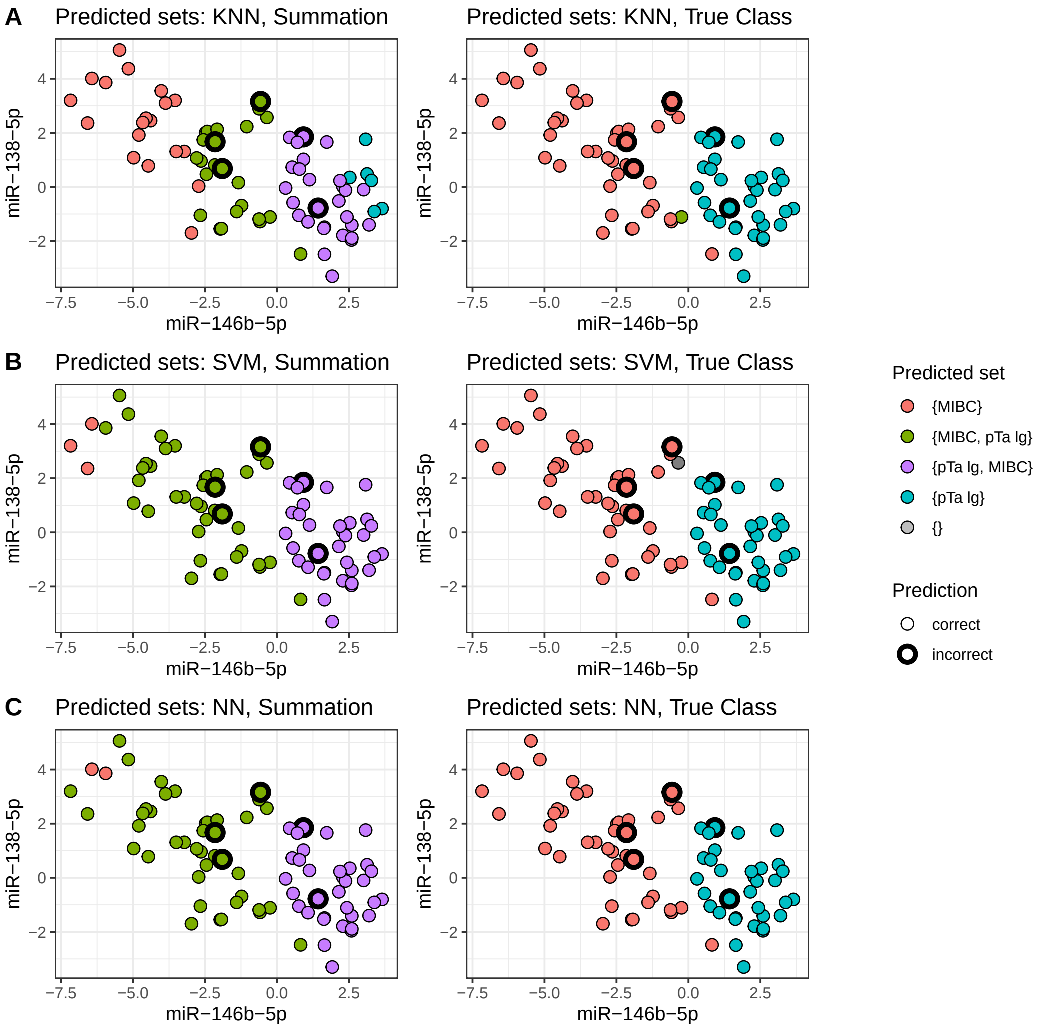


Figure G.9: Comparison of Summation and True Class score. This figure compares the predicted sets of the best-performing k-nearest neighbors model (A), support vector machine (B), and vanilla neural network (C) using the Summation (left side) vs. the True Class score (right side). In each plot, the expression of miR-146b-5p and miR-138-5p is shown as ΔCP values and each point corresponds to one test sample.

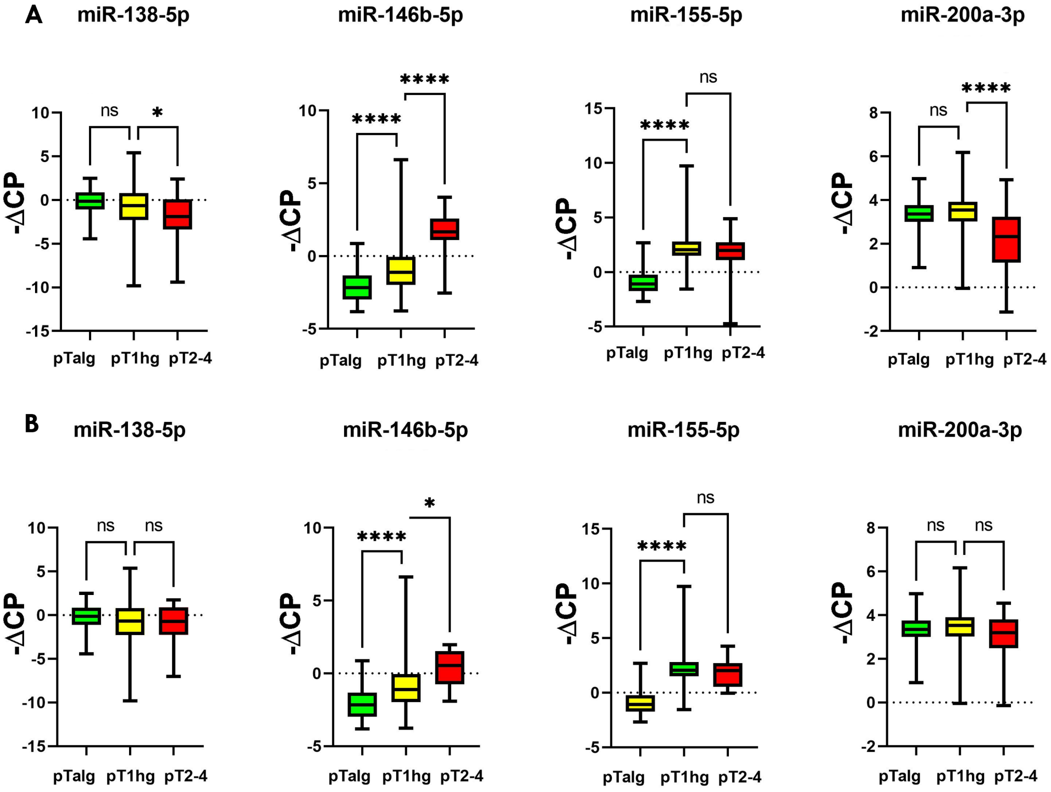


Figure G.10: Expression of miRNAs in CYS-MIBC (A) and TUR-MIBC (B) vs. pTa lg and pT1 hg cancers in Cohort 2. For each miRNA, its expression is shown as $-\Delta$CP values for the MIBC (denoted as pT2-4), pT1 hg, and pTa lg samples. Significant expression differences between the groups were identified using Mann-Whitney U tests. The corresponding p-values (p) are annotated as $*$ for $p \leq 0.05$, as $****$ for $p \leq 0.0001$, and as *ns* for $p > 0.05$.

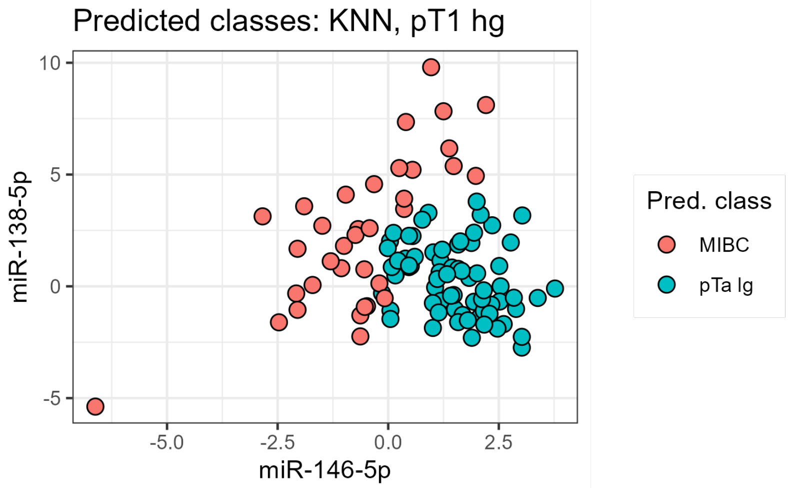


Figure G.11: Expression of miR-146-5p and miR-138-5p (shown as ΔCP values) for the pT1 hg samples of Cohort 2. Points are colored according to the class prediction of the best-performing KNN model.

Bibliography

[1] A. Sudhakar. “History of cancer, ancient and modern treatment methods”. In: *Journal of cancer science & therapy* 1.2 (2009), p. 1.

[2] F. Bray, M. Laversanne, H. Sung, J. Ferlay, R. L. Siegel, I. Soerjomataram, and A. Jemal. “Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries”. In: *CA: a cancer journal for clinicians* 74.3 (2024), pp. 229–263. DOI: 10.3322/caac.21834.

[3] C. Frick, H. Rumgay, J. Vignat, O. Ginsburg, E. Nolte, F. Bray, and I. Soerjomataram. “Quantitative estimates of preventable and treatable deaths from 36 cancers worldwide: a population-based study”. In: *The Lancet Global Health* 11.11 (2023), e1700–e1712. DOI: 10.1016/s2214-109x(23)00406-0.

[4] K. B. Tran, J. J. Lang, K. Compton, R. Xu, A. R. Acheson, H. J. Henrikson, J. M. Kocarnik, L. Penberthy, A. Aali, Q. Abbas, et al. “The global burden of cancer attributable to risk factors, 2010–19: a systematic analysis for the Global Burden of Disease Study 2019”. In: *The Lancet* 400.10352 (2022), pp. 563–591. DOI: 10.1016/S0140-6736(22)01438-6.

[5] V. A. Katzke, R. Kaaks, and T. Kühn. “Lifestyle and cancer risk”. In: *The Cancer Journal* 21.2 (2015), pp. 104–110. DOI: 10.1097/ppo.0000000000000101.

[6] P. Boffetta and F. Nyberg. “Contribution of environmental factors to cancer risk”. In: *British medical bulletin* 68.1 (2003), pp. 71–94. DOI: 10.1093/bmp/ldg023.

[7] D. L. Narayanan, R. N. Saladi, and J. L. Fox. “Ultraviolet radiation and skin cancer”. In: *International journal of dermatology* 49.9 (2010), pp. 978–986. DOI: 10.1111/j.1365-4632.2010.04474.x.

[8] A. G. Knudson. “Hereditary predisposition to cancer”. In: *Annals of the New York Academy of Sciences* 833.1 (1997), pp. 58–67. DOI: 10.1111/j.1749-6632.1997.tb48593.x.

[9] B. Bottazzi, E. Riboli, and A. Mantovani. “Aging, inflammation and cancer”. In: *Seminars in immunology*. Vol. 40. Elsevier. 2018, pp. 74–82. DOI: 10.1016/j.smim.2018.10.011.

[10] A. Urruticoechea, R. Alemany, J Balart, A. Villanueva, F. Vinals, and G. Capella. "Recent advances in cancer therapy: an overview". In: *Current pharmaceutical design* 16.1 (2010), pp. 3–10. DOI: 10.2174/138161210789941847.

[11] E. Pérez-Herrero and A. Fernández-Medarde. "Advanced targeted therapies in cancer: Drug nanocarriers, the future of chemotherapy". In: *European journal of pharmaceutics and biopharmaceutics* 93 (2015), pp. 52–79. DOI: 10.1016/j.ejpb.2015.03.018.

[12] I. Altun and A. Sonkaya. "The most common side effects experienced by patients were receiving first cycle of chemotherapy". In: *Iranian journal of public health* 47.8 (2018), pp. 1218–1219.

[13] A. Sourati, A. Ameri, and M. Malekzadeh. "Acute side effects of radiation therapy". In: *Cham: Springer* (2017). DOI: 10.1007/978-3-319-55950-6.

[14] J. Zeien, W. Qiu, M. Triay, H. A. Dhaibar, D. Cruz-Topete, E. M. Cornett, I. Urits, O. Viswanath, and A. D. Kaye. "Clinical implications of chemotherapeutic agent organ toxicity on perioperative care". In: *Biomedicine & Pharmacotherapy* 146 (2022), p. 112503. DOI: 10.1016/j.biopha.2021.112503.

[15] M. Kosmin and J. Rees. "Radiation and the nervous system". In: *Practical Neurology* 22.6 (2022), pp. 450–460. DOI: 10.1136/pn-2022-003343.

[16] D. Rasnick. "Aneuploidy theory explains tumor formation, the absence of immune surveillance, and the failure of chemotherapy". In: *Cancer genetics and cytogenetics* 136.1 (2002), pp. 66–72. DOI: 10.1016/s0165-4608(01)00665-3.

[17] R. Nahta and F. Esteva. "Trastuzumab: triumphs and tribulations". In: *Oncogene* 26.25 (2007), pp. 3637–3643. DOI: 10.1038/sj.onc.1210379.

[18] M. Huang, A. Shen, J. Ding, and M. Geng. "Molecularly targeted cancer therapy: some lessons from the past decade". In: *Trends in pharmacological sciences* 35.1 (2014), pp. 41–50. DOI: 10.1016/j.tips.2013.11.004.

[19] S Garattini, I. F. Nerini, and M D'incalci. "Not only tumor but also therapy heterogeneity". In: *Annals of Oncology* 29.1 (2018), pp. 13–18. DOI: 10.1093/annonc/mdx646.

[20] E. Abrahams and M. Silver. "The history of personalized medicine". In: *Integrative neuroscience and personalized medicine* (2010), pp. 3–16. DOI: 10.1093/acprof:oso/9780195393804.003.0001.

[21] A. Goodspeed, L. M. Heiser, J. W. Gray, and J. C. Costello. "Tumor-derived cell lines as molecular models of cancer pharmacogenomics". In: *Molecular Cancer Research* 14.1 (2016), pp. 3–13. DOI: 10.1158/1541-7786.mcr-15-0189.

[22] E. A. Brooks, S. Galarza, M. F. Gencoglu, R. C. Cornelison, J. M. Munson, and S. R. Peyton. "Applicability of drug response metrics for cancer studies using biomaterials". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 374.1779 (2019), p. 20180226. DOI: 10.1098/rstb.2018.0226.

[23] I. S. Jang, E. C. Neto, J. Guinney, S. H. Friend, and A. A. Margolin. "Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data". In: *Biocomputing 2014*. World Scientific, 2014, pp. 63–74. DOI: 10.1142/9789814583220_0007.

[24] Y. Li, D. E. Hostallero, and A. Emad. "Interpretable deep learning architectures for improving drug response prediction performance: myth or reality?" In: *Bioinformatics* 39.6 (2023), btad390. DOI: 10.1093/bioinformatics/btad390.

[25] Y. Chen and L. Zhang. "How much can deep learning improve prediction of the responses to drugs in cancer cell lines?" In: *Briefings in bioinformatics* 23.1 (2022), bbab378. DOI: 10.1093/bib/bbab378.

[26] K. Lenhof, L. Eckhart, L.-M. Rolli, and H.-P. Lenhof. "Trust me if you can: a survey on reliability and interpretability of machine learning approaches for drug sensitivity prediction in cancer". In: *Briefings in Bioinformatics* 25.5 (Aug. 2024), bbae379. DOI: 10.1093/bib/bbae379.

[27] M. Kukar and I. Kononenko. "Reliable classifications with machine learning". In: *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*. Springer. 2002, pp. 219–231. DOI: 10.1007/3-540-36755-1_19.

[28] V.-L. Nguyen, S. Destercke, M.-H. Masson, and E. Hüllermeier. "Reliable multiclass classification based on pairwise epistemic and aleatoric uncertainty". In: *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. 2018, pp. 5089–5095. DOI: 10.24963/ijcai.2018/706.

[29] G. Nicora, M. Rios, A. Abu-Hanna, and R. Bellazzi. "Evaluating pointwise reliability of machine learning prediction". In: *Journal of Biomedical Informatics* 127 (2022), p. 103996. DOI: 10.1016/j.jbi.2022.103996.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[31] F. Imrie, R. Davis, and M. van der Schaar. "Multiple stakeholders drive diverse interpretability requirements for machine learning in healthcare". In: *Nature Machine Intelligence* 5 (2023), 824–829. DOI: 10.1038/s42256-023-00698-2.

[32] O. Biran and C. Cotton. "Explanation and justification in machine learning: A survey". In: *IJCAI-17 workshop on explainable AI (XAI)*. Vol. 8. 1. 2017, pp. 8–13.

[33] D. Baptista, P. G. Ferreira, and M. Rocha. "Deep learning for drug response prediction in cancer". In: *Briefings in Bioinformatics* 22.1 (Jan. 2020), pp. 360–379. ISSN: 1477-4054. DOI: 10.1093/bib/bbz171.

[34] X. An, X. Chen, D. Yi, H. Li, and Y. Guan. "Representation of molecules for drug response prediction". In: *Briefings in Bioinformatics* 23.1 (Sept. 2021), bbab393. ISSN: 1477-4054. DOI: 10.1093/bib/bbab393.

[35] L. Eckhart, K. Lenhof, L.-M. Rolli, and H.-P. Lenhof. "A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction". In: *Briefings in bioinformatics* 25.5 (2024). DOI: 10.1093/bib/bbae242.

[36] T. A. Knijnenburg, G. W. Klau, F. Iorio, M. J. Garnett, U. McDermott, I. Shmulevich, and L. F. Wessels. "Logic models to predict continuous outputs based on binary inputs with an application to personalized cancer therapy". In: *Scientific reports* 6.1 (2016), pp. 1–14. DOI: 10.1038/srep36812.

[37] K. Lenhof, N. Gerstner, T. Kehl, L. Eckhart, L. Schneider, and H.-P. Lenhof. "MERIDA: a novel Boolean logic-based integer linear program for personalized cancer therapy". In: *Bioinformatics* 37.21 (2021), pp. 3881–3888. DOI: 10.1093/bioinformatics/btab546.

[38] A. Basu, R. Mitra, H. Liu, S. L. Schreiber, and P. A. Clemons. "RWEN: response-weighted elastic net for prediction of chemosensitivity of cancer cell lines". In: *Bioinformatics* 34.19 (2018), pp. 3332–3339. DOI: 10.1093/bioinformatics/bty199.

[39] K. Lenhof, L. Eckhart, N. Gerstner, T. Kehl, and H.-P. Lenhof. "Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method". In: *Scientific Reports* 12.1 (2022), p. 13458. DOI: 10.1038/s41598-022-17609-x.

[40] K. Lenhof, L. Eckhart, L.-M. Rolli, A. Volkamer, and H.-P. Lenhof. "Reliable anti-cancer drug sensitivity prediction and prioritization". In: *Scientific Reports* 14.1 (2024), p. 12303. DOI: 10.1038/s41598-024-62956-6.

[41] R. B. Mokhtari, T. S. Homayouni, N. Baluch, E. Morgatskaya, S. Kumar, B. Das, and H. Yeger. "Combination therapy in combating cancer". In: *Oncotarget* 8.23 (2017), p. 38022. DOI: 10.18632%2Foncotarget.16723.

[42] L. Wu, Y. Wen, D. Leng, Q. Zhang, C. Dai, Z. Wang, Z. Liu, B. Yan, Y. Zhang, J. Wang, et al. "Machine learning methods, databases and tools for drug combination prediction". In: *Briefings in bioinformatics* 23.1 (2022), bbab355. DOI: 10.1093/bib/bbab355.

[43] A. H. Vlot, N. Aniceto, M. P. Menden, G. Ulrich-Merzenich, and A. Bender. "Applying synergy metrics to combination screening data: agreements, disagreements and pitfalls". In: *Drug discovery today* 24.12 (2019), pp. 2286–2298. DOI: 10.1016/j.drudis.2019.09.002.

[44] S. Lederer, T. M. Dijkstra, and T. Heskes. "Additive dose response models: explicit formulation and the loewe additivity consistency condition". In: *Frontiers in pharmacology* 9 (2018), p. 31. DOI: 10.3389/fphar.2018.00031.

[45] W. R. Greco, G Bravo, and J. C. Parsons. "The search for synergy: a critical review from a response surface perspective." In: *Pharmacological Reviews* 47.2 (1995), pp. 331–385. ISSN: 0031-6997. URL: https://pharmrev.aspetjournals.org/content/47/2/331.

[46] S. Zheng, J. Aldahdooh, T. Shadbahr, Y. Wang, D. Aldahdooh, J. Bao, W. Wang, and J. Tang. "DrugComb update: a more comprehensive drug sensitivity data repository and analysis portal". In: *Nucleic acids research* 49.W1 (2021), W174–W184. DOI: 10.1093%2Fnar%2Fgkab438.

[47] L. Eckhart, K. Lenhof, L. Herrmann, L.-M. Rolli, and H.-P. Lenhof. "How to Predict Effective Drug Combinations-Moving beyond Synergy Scores". In: *bioRxiv* (2024), pp. 2024–11. DOI: 10.1101/2024.11.22.624812.

[48] L. Eckhart, S. Rau, M. Eckstein, P. R. Stahl, H. Ayoubian, J. Heinzelbecker, F. Zohari, A. Hartmann, M. Stöckle, H.-P. Lenhof, et al. "Machine Learning Accurately Predicts Muscle Invasion of Bladder Cancer Based on Three miRNAs". In: *Journal of Cellular and Molecular Medicine* 29.3 (2025), e70361. DOI: 10.1111/jcmm.70361.

[49] F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot, et al. "A landscape of pharmacogenomic interactions in cancer". In: *Cell* 166.3 (2016), pp. 740–754. DOI: 10.1016/j.cell.2016.06.017.

[50] B. Zagidullin, J. Aldahdooh, S. Zheng, W. Wang, Y. Wang, J. Saad, A. Malyutina, M. Jafari, Z. Tanoli, A. Pessia, et al. "DrugComb: an integrative cancer drug combination data portal". In: *Nucleic acids research* 47.W1 (2019), W43–W51. DOI: 10.1093/nar/gkz337.

[51] E. J. Odes, P. S. Randolph-Quinney, M. Steyn, Z. Throckmorton, J. S. Smilg, B. Zipfel, T. N. Augustine, F. De Beer, J. W. Hoffman, R. D. Franklin, et al. "Earliest hominin cancer: 1.7-million-year-old osteosarcoma from Swartkrans Cave, South Africa". In: *South African Journal of Science* 112.7-8 (2016), pp. 1–5. DOI: 10.17159/sajs.2016/20150471.

[52] J. P. Allen. *The art of medicine in ancient Egypt.* Metropolitan Museum of Art, 2005.

[53] G. Tsoucalas and M. Sgantzos. "Hippocrates (ca 460-370 BC) on nasal cancer". In: *Cancer* 4 (2016), p. 5.

[54] A. Di Lonardo, S. Nasi, and S. Pulciani. "Cancer: we should not forget the past". In: *Journal of cancer* 6.1 (2015), p. 29. DOI: 10.7150/jca.10336.

[55] R. P. Wagner. "Rudolph Virchow and the genetic basis of somatic ecology". In: *Genetics* 151.3 (1999), pp. 917–920. DOI: 10.1093/genetics/151.3.917.

[56] J. R. Brown and J. L. Thornton. "Percivall Pott (1714-1788) and chimney sweepers' cancer of the scrotum". In: *British journal of industrial medicine* 14.1 (1957), p. 68. DOI: 10.1136/oem.14.1.68.

[57] V. T. DeVita Jr and S. A. Rosenberg. "Two hundred years of cancer research". In: *New England Journal of Medicine* 366.23 (2012), pp. 2207–2214. DOI: 10.1056/NEJMra1204479.

[58] C. R. Hayter. "The clinic as laboratory: The case of radiation therapy, 1896-1920". In: *Bulletin of the History of Medicine* 72.4 (1998), pp. 663–688. DOI: 10.1353/bhm.1998.0213.

[59] V. T. DeVita Jr and E. Chu. "A history of cancer chemotherapy". In: *Cancer research* 68.21 (2008), pp. 8643–8653. DOI: 10.1158/0008-5472.CAN-07-6611.

[60] B. Alberts, R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell.* 7th. New York: W. W. Norton & Company, 2022. ISBN: 9780393884821.

[61] L. Eckhart. *Flow of genetic information.* Created in BioRender. 2025. URL: https://BioRender.com/r57m466 (visited on 05/11/2025).

[62] T. Phillips and K. Shaw. "Chromatin remodeling in eukaryotes". In: *Nature Education* 1.1 (2008), p. 209.

[63] A. K. Maunakea, I. Chepelev, and K. Zhao. "Epigenome mapping in normal and disease States". In: *Circulation research* 107.3 (2010), pp. 327–339. DOI: 10.1161/CIRCRESAHA.110.222463.

[64] R. Lowe, N. Shirley, M. Bleackley, S. Dolan, and T. Shafee. “Transcriptomics technologies”. In: *PLoS computational biology* 13.5 (2017), e1005457. DOI: 10.1371/journal.pcbi.1005457.

[65] J. Macedo-da Silva, C. R. F. Marinho, G. Palmisano, and L. Rosa-Fernandes. “Lights and shadows of TORCH infection proteomics”. In: *Genes* 11.8 (2020), p. 894. DOI: 10.3390/genes11080894.

[66] H. Khatter, A. G. Myasnikov, S. K. Natchiar, and B. P. Klaholz. “Structure of the human 80S ribosome”. In: *Nature* 520.7549 (2015), pp. 640–645. DOI: 10.1038/nature14427.

[67] D. A. Jans and S. Hubner. “Regulation of protein transport to the nucleus: central role of phosphorylation”. In: *Physiological reviews* 76.3 (1996), pp. 651–685. DOI: 10.1152/physrev.1996.76.3.651.

[68] J. F. Seidler and K. Sträßer. “Understanding nuclear mRNA export: Survival under stress”. In: *Molecular Cell* 84.19 (2024), pp. 3681–3691. DOI: 10.1016/j.molcel.2024.08.028.

[69] J. Wang, M. Horlacher, L. Cheng, and O. Winther. “RNA trafficking and subcellular localization—a review of mechanisms, experimental and predictive methodologies”. In: *Briefings in bioinformatics* 24.5 (2023), bbad249. DOI: 10.1093/bib/bbad249.

[70] J. Alles, T. Fehlmann, U. Fischer, C. Backes, V. Galata, M. Minet, M. Hart, M. Abu-Halima, F. A. Grässer, H.-P. Lenhof, et al. “An estimate of the total number of true human miRNAs”. In: *Nucleic acids research* 47.7 (2019), pp. 3353–3364. DOI: 10.1093/nar/gkz097.

[71] S. Chatterjee and N. Ahituv. “Gene regulatory elements, major drivers of human disease”. In: *Annual review of genomics and human genetics* 18.1 (2017), pp. 45–63. DOI: 10.1146/annurev-genom-091416-035537.

[72] A. H. Corbett. “Post-transcriptional regulation of gene expression and human disease”. In: *Current opinion in cell biology* 52 (2018), pp. 96–104. DOI: 10.1016/j.ceb.2018.02.011.

[73] M. Gos. “Epigenetic mechanisms of gene expression regulation in neurological diseases”. In: *Acta neurobiologiae experimentalis* 73.1 (2013), pp. 19–37. DOI: 10.55782/ane-2013-1919.

[74] R. Lewis. *Human Genetics: Concepts and Applications*. 9th. McGraw-Hill Primis, 2016. ISBN: 9780390232441.

[75] National Human Genome Research Institute. *Fact Sheet: Human Genomic Variation*. 2023. URL: https://www.genome.gov/about-genomics/educational-resources/fact-sheets/human-genomic-variation (visited on 05/11/2025).

[76] B. Alberts, R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, and P. Walter. "Molecular Biology of the Cell". In: 7th. New York: W. W. Norton & Company, 2022. Chap. 1. ISBN: 9780393884821.

[77] J. Graw. "Genetik". In: 7th. 7th edition. Heidelberg: Springer Spektrum Berlin, 2020. Chap. 10.1. ISBN: 9783662609095. DOI: 10.1007/978-3-662-60909-5.

[78] C. Melton, J. A. Reuter, D. V. Spacek, and M. Snyder. "Recurrent somatic mutations in regulatory regions of human cancer genomes". In: *Nature genetics* 47.7 (2015), pp. 710–716. DOI: 10.1038/ng.3332.

[79] B. Alberts, R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, and P. Walter. "Molecular Biology of the Cell". In: 7th. New York: W. W. Norton & Company, 2022. Chap. 6. ISBN: 9780393884821.

[80] C. Pagiatakis, E. Musolino, R. Gornati, G. Bernardini, and R. Papait. "Epigenetics of aging and disease: a brief overview". In: *Aging clinical and experimental research* 33 (2021), pp. 737–745. DOI: 10.1007/s40520-019-01430-0.

[81] A. V. Probst, E. Dunleavy, and G. Almouzni. "Epigenetic inheritance during the cell cycle". In: *Nature reviews Molecular cell biology* 10.3 (2009), pp. 192–206. DOI: 10.1038/nrm2640.

[82] B. Alberts, R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, and P. Walter. "Molecular Biology of the Cell". In: 7th. New York: W. W. Norton & Company, 2022. Chap. 5. ISBN: 9780393884821.

[83] B. Alberts, R. Heald, A. Johnson, D. Morgan, M. Raff, K. Roberts, and P. Walter. "Molecular Biology of the Cell". In: 7th. New York: W. W. Norton & Company, 2022. Chap. 4. ISBN: 9780393884821.

[84] G. Gibson. "Decanalization and the origin of complex disease". In: *Nature Reviews Genetics* 10.2 (2009), pp. 134–140. DOI: 10.1038/nrg2502.

[85] K. J. Mitchell. "What is complex about complex disorders?" In: *Genome biology* 13 (2012), pp. 1–11. DOI: 10.1186/gb-2012-13-1-237.

[86] German Federal Statistical Office (Destatis). *Todesursachen 2022: Anteil der an COVID-19-Verstorbenen rückläufig*. 2022. URL: https://www.destatis.de/DE/Presse/Pressemitteilungen/2023/11/PD23_441_23211.html (visited on 05/11/2025).

[87] M. R. Stratton, P. J. Campbell, and P. A. Futreal. "The cancer genome". In: *Nature* 458.7239 (2009), pp. 719–724. DOI: 10.1038/nature07943.

[88] M. Sinkala. “Mutational landscape of cancer-driver genes across human cancers”. In: *Scientific Reports* 13.1 (2023), p. 12742. DOI: 10.1038/s41598-023-39608-2.

[89] E. Y. Lee and W. J. Muller. “Oncogenes and tumor suppressor genes”. In: *Cold Spring Harbor perspectives in biology* 2.10 (2010), a003236. DOI: 10.1101/cshperspect.a003236.

[90] D. Hanahan and R. A. Weinberg. “The hallmarks of cancer”. In: *cell* 100.1 (2000), pp. 57–70. DOI: 10.1016/s0092-8674(00)81683-9.

[91] D. Hanahan and R. A. Weinberg. “Hallmarks of cancer: the next generation”. In: *cell* 144.5 (2011), pp. 646–674. DOI: 10.1016/j.cell.2011.02.013.

[92] D. Hanahan. “Hallmarks of cancer: new dimensions”. In: *Cancer discovery* 12.1 (2022), pp. 31–46. DOI: 10.1158/2159-8290.cd-21-1059.

[93] C. Criscitiello. “Tumor-associated antigens in breast cancer”. In: *Breast care* 7.4 (2012), pp. 262–266. DOI: 10.1159/000342164.

[94] M. De Charette, A. Marabelle, and R. Houot. “Turning tumour cells into antigen presenting cells: The next step to improve cancer immunotherapy?” In: *European Journal of Cancer* 68 (2016), pp. 134–147. DOI: 10.1016/j.ejca.2016.09.010.

[95] D. M. Gress, S. B. Edge, F. L. Greene, M. K. Washington, E. A. Asare, J. D. Brierley, D. R. Byrd, C. C. Compton, J. M. Jessup, D. P. Winchester, et al. “Principles of cancer staging”. In: *AJCC cancer staging manual* 8 (2017), pp. 3–30.

[96] J. D. Brierley, M. K. Gospodarowicz, and C. Wittekind. *TNM classification of malignant tumours*. John Wiley & Sons, 2017. ISBN: 9781119263562.

[97] A. Carbone. “Cancer classification at the crossroads”. In: *Cancers* 12.4 (2020), p. 980. DOI: 10.3390/cancers12040980.

[98] E. Orrantia-Borunda, P. Anchondo-Nuñez, L. E. Acuña-Aguilar, F. O. Gómez-Valles, and C. A. Ramírez-Valdespino. “Chapter 3: Subtypes of breast cancer”. In: *Breast Cancer [Internet]* (2022). DOI: 10.36255/exon-publications-breast-cancer-subtypes.

[99] R. Baskar, K. A. Lee, R. Yeo, and K.-W. Yeoh. “Cancer and radiation therapy: current advances and future directions”. In: *International journal of medical sciences* 9.3 (2012), p. 193. DOI: 10.7150/ijms.3635.

[100] J. Fernando and R. Jones. “The principles of cancer treatment by chemotherapy”. In: *Surgery (Oxford)* 33.3 (2015), pp. 131–135. DOI: 10.1016/j.mpsur.2015.01.005.

[101] H. Jin, L. Wang, and R. Bernards. “Rational combinations of targeted cancer therapies: background, advances and challenges”. In: *Nature Reviews Drug Discovery* 22.3 (2023), pp. 213–234. DOI: 10.1038/s41573-022-00615-z.

[102] A. H. Sharpe. “Introduction to checkpoint inhibitors and cancer immunotherapy”. In: *Immunological reviews* 276.1 (2017), p. 5. DOI: 10.1111/imr.12531.

[103] M. J. Lin, J. Svensson-Arvelund, G. S. Lubitz, A. Marabelle, I. Melero, B. D. Brown, and J. D. Brody. “Cancer vaccines: the next immunotherapy frontier”. In: *Nature cancer* 3.8 (2022), pp. 911–926. DOI: 10.1038/s43018-022-00418-6.

[104] T. Haslauer, R. Greil, N. Zaborsky, and R. Geisberger. “CAR T-cell therapy in hematological malignancies”. In: *International Journal of Molecular Sciences* 22.16 (2021), p. 8996. DOI: 10.3390/ijms22168996.

[105] M. Rucińska. “Combined radiotherapy and chemotherapy”. In: *Nowotwory. Journal of Oncology* 72.5 (2022), pp. 319–325. DOI: 10.5603/NJO.2022.0051.

[106] A. Kalbasi, C. H. June, N. Haas, N. Vapiwala, et al. “Radiation and immunotherapy: a synergistic combination”. In: *The Journal of clinical investigation* 123.7 (2013), pp. 2756–2763. DOI: 10.1172/JCI69219.

[107] M. Vanneman and G. Dranoff. “Combining immunotherapy and targeted therapies in cancer treatment”. In: *Nature reviews cancer* 12.4 (2012), pp. 237–251. DOI: 10.1038/nrc3237.

[108] S. Fudio, A. Sellers, L. Pérez Ramos, B. Gil-Alberdi, A. Zeaiter, M. Urroz, A. Carcas, and R. Lubomirov. “Anti-cancer drug combinations approved by US FDA from 2011 to 2021: main design features of clinical trials and role of pharmacokinetics”. In: *Cancer Chemotherapy and Pharmacology* 90.4 (2022), pp. 285–299. DOI: 10.1007/s00280-022-04467-7.

[109] K. Swanson, E. Wu, A. Zhang, A. A. Alizadeh, and J. Zou. “From patterns to patients: Advances in clinical machine learning for cancer diagnosis, prognosis, and treatment”. In: *Cell* 186.8 (2023), pp. 1772–1791. DOI: 10.1016/j.cell.2023.01.035.

[110] M. J. Iqbal, Z. Javed, H. Sadia, I. A. Qureshi, A. Irshad, R. Ahmed, K. Malik, S. Raza, A. Abbas, R. Pezzani, et al. “Clinical applications of artificial intelligence and machine learning in cancer diagnosis: looking into the future”. In: *Cancer cell international* 21.1 (2021), p. 270. DOI: 10.1186/s12935-021-01981-1.

[111] S. Albaradei, M. Thafar, A. Alsaedi, C. Van Neste, T. Gojobori, M. Essack, and X. Gao. “Machine learning and deep learning methods that use omics data for metastasis prediction”. In: *Computational and structural biotechnology journal* 19 (2021), pp. 5008–5018. DOI: 10.1016/j.csbj.2021.09.001.

[112] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson. "A survey on graph-based deep learning for computational histopathology". In: *Computerized Medical Imaging and Graphics* 95 (2022), p. 102027. DOI: 10.1016/j.compmedimag.2021.102027.

[113] P Deepa and C Gunavathi. "A systematic review on machine learning and deep learning techniques in cancer survival prediction". In: *Progress in Biophysics and Molecular Biology* 174 (2022), pp. 62–71. DOI: 10.1016/j.pbiomolbio.2022.07.004.

[114] H. Zhao, J. Zhong, X. Liang, C. Xie, and S. Wang. "Application of machine learning in drug side effect prediction: databases, methods, and challenges". In: *Frontiers of Computer Science* 19.5 (2025), p. 195902. DOI: 10.1007/s11704-024-31063-0.

[115] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, et al. "Applications of machine learning in drug discovery and development". In: *Nature reviews Drug discovery* 18.6 (2019), pp. 463–477. DOI: 10.1038/s41573-019-0024-5.

[116] L. Eckhart. "Drug Sensitivity Prediction Using Neural Networks". Master's thesis. Saarland University, 2020.

[117] M. A. Barbosa, C. P. Xavier, R. F. Pereira, V. Petrikaitė, and M. H. Vasconcelos. "3D cell culture models as recapitulators of the tumor microenvironment for the screening of anti-cancer drugs". In: *Cancers* 14.1 (2021), p. 190. DOI: 10.3390/cancers14010190.

[118] S. Breslin and L. O'Driscoll. "Three-dimensional cell culture: the missing link in drug discovery". In: *Drug discovery today* 18.5-6 (2013), pp. 240–249. DOI: 10.1016/j.drudis.2012.10.003.

[119] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, et al. "The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity". In: *Nature* 483.7391 (2012), pp. 603–607. DOI: 10.1016/s0959-8049(12)70726-8.

[120] M. L. Sos, K. Michel, T. Zander, J. Weiss, P. Frommolt, M. Peifer, D. Li, R. Ullrich, M. Koker, F. Fischer, et al. "Predicting drug susceptibility of non–small cell lung cancers based on genetic lesions". In: *The Journal of clinical investigation* 119.6 (2009), pp. 1727–1740. DOI: 10.1172/jci37127.

[121] R. M. Neve, K. Chin, J. Fridlyand, J. Yeh, F. L. Baehner, T. Fevr, L. Clark, N. Bayani, J.-P. Coppe, F. Tong, et al. "A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes". In: *Cancer cell* 10.6 (2006), pp. 515–527. DOI: 10.1016/j.ccr.2006.10.008.

[122] R. H. Shoemaker. “The NCI60 human tumour cell line anticancer drug screen”. In: *Nature Reviews Cancer* 6.10 (2006), pp. 813–823. DOI: 10.1038/nrc1951.

[123] K Bracht, A. Nicholls, Y Liu, and W. Bodmer. “5-Fluorouracil response in a large panel of colorectal cancer cell lines is associated with mismatch repair deficiency”. In: *British journal of cancer* 103.3 (2010), pp. 340–346. DOI: 10.1038/sj.bjc.6605780.

[124] G. Jiang, S. Zhang, A. Yazdanparast, M. Li, A. V. Pawar, Y. Liu, S. M. Inavolu, and L. Cheng. “Comprehensive comparison of molecular portraits between cell lines and tumors in breast cancer”. In: *BMC genomics* 17 (2016), pp. 281–301. DOI: 10.1186/s12864-016-2911-z.

[125] T. Voskoglou-Nomikos, J. L. Pater, and L. Seymour. “Clinical predictive value of the in vitro cell line, human xenograft, and mouse allograft preclinical cancer models”. In: *Clinical cancer research* 9.11 (2003), pp. 4227–4239.

[126] G. Mazzoleni, D Di Lorenzo, and N. Steimberg. “Modelling tissues in 3D: the next future of pharmaco-toxicology and food research?” In: *Genes & nutrition* 4 (2009), pp. 13–22. DOI: 10.1007/s12263-008-0107-0.

[127] A. A. Rizvanov, M. E. Yalvaç, A. K. Shafigullina, I. I. Salafutdinov, N. L. Blatt, F. Sahin, A. P. Kiyasov, and A. Palotás. “Interaction and self-organization of human mesenchymal stem cells and neuro-blastoma SH-SY5Y cells under co-culture conditions: A novel system for modeling cancer cell micro-environment”. In: *European journal of pharmaceutics and biopharmaceutics* 76.2 (2010), pp. 253–259. DOI: 10.1016/j.ejpb.2010.05.012.

[128] J. L. Wilding and W. F. Bodmer. “Cancer cell lines for drug discovery and development”. In: *Cancer research* 74.9 (2014), pp. 2377–2384. DOI: 10.1158/0008-5472.can-13-2971.

[129] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, et al. “Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells”. In: *Nucleic acids research* 41.D1 (2012), pp. D955–D961. DOI: 10.1093/nar/gks1111.

[130] Wellcome Sanger Institute, GDSC database. *Help and Documentation - Screening*. 2019. URL: https://www.cancerrxgene.org/help#t_curve (visited on 05/11/2025).

[131] A. Basu, N. E. Bodycombe, J. H. Cheah, E. V. Price, K. Liu, G. I. Schaefer, R. Y. Ebright, M. L. Stewart, D. Ito, S. Wang, et al. “An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules”. In: *Cell* 154.5 (2013), pp. 1151–1161. DOI: 10.1016/j.cell.2013.08.003.

[132] B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, N. E. Bodycombe, C. K. Soule, J. Gould, et al. "Harnessing connectivity in a large-scale small-molecule sensitivity dataset". In: *Cancer discovery* 5.11 (2015), pp. 1210–1223. DOI: 10.1158/2159-8290.CD-15-0235.

[133] M. G. Rees, B. Seashore-Ludlow, J. H. Cheah, D. J. Adams, E. V. Price, S. Gill, S. Javaid, M. E. Coletti, V. L. Jones, N. E. Bodycombe, C. K. Soule, B. Alexander, A. Li, P. Montgomery, J. D. Kotz, C. S.-Y. Hon, B. Munoz, T. Liefeld, V. Dančík, D. A. Haber, C. B. Clish, J. A. Bittker, M. Palmer, B. K. Wagner, P. A. Clemons, A. F. Shamji, and S. L. Schreiber. "Correlating chemical sensitivity and basal gene expression reveals mechanism of action". In: *Nature Chemical Biology* 12.2 (2015), pp. 109–116. DOI: 10.1038/nchembio.1986.

[134] S. M. Corsello, R. T. Nagari, R. D. Spangler, J. Rossen, M. Kocak, J. G. Bryan, R. Humeidi, D. Peck, X. Wu, A. A. Tang, et al. "Discovering the anticancer potential of non-oncology drugs by systematic viability profiling". In: *Nature cancer* 1.2 (2020), pp. 235–248. DOI: 10.1038/s43018-019-0018-6.

[135] P. M. Haverty, E. Lin, J. Tan, Y. Yu, B. Lam, S. Lianoglou, R. M. Neve, S. Martin, J. Settleman, R. L. Yauch, et al. "Reproducible pharmacogenomic profiling of cancer cell line panels". In: *Nature* 533.7603 (2016), pp. 333–337. DOI: 10.1038/nature17987.

[136] C. Klijn, S. Durinck, E. W. Stawiski, P. M. Haverty, Z. Jiang, H. Liu, J. Degenhardt, O. Mayba, F. Gnad, J. Liu, et al. "A comprehensive transcriptional portrait of human cancer cell lines". In: *Nature biotechnology* 33.3 (2015), pp. 306–312. DOI: 10.1038/nbt.3080.

[137] J. Greshock, K. E. Bachman, Y. Y. Degenhardt, J. Jing, Y. H. Wen, S. Eastman, E. McNeil, C. Moy, R. Wegrzyn, K. Auger, et al. "Molecular target class is predictive of in vitro response profile". In: *Cancer research* 70.9 (2010), pp. 3677–3686. DOI: 10.1158/0008-5472.can-09-3788.

[138] J. P. Mpindi, B. Yadav, P. Östling, P. Gautam, D. Malani, A. Murumägi, A. Hirasawa, S. Kangaspeska, K. Wennerberg, O. Kallioniemi, et al. "Consistency in drug response profiling". In: *Nature* 540.7631 (2016), E5–E6. DOI: 10.1038/nature20171.

[139] S. L. Holbeck, R. Camalier, J. A. Crowell, J. P. Govindharajulu, M. Hollingshead, L. W. Anderson, E. Polley, L. Rubinstein, A. Srivastava, D. Wilsker, et al. "The National Cancer Institute ALMANAC: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity". In: *Cancer research* 77.13 (2017), pp. 3564–3576. DOI: 10.1158/0008-5472.can-17-0489.

[140] J. O'Neil, Y. Benita, I. Feldman, M. Chenard, B. Roberts, Y. Liu, J. Li, A. Kral, S. Lejnine, A. Loboda, et al. "An unbiased oncology compound screen to identify novel combination strategies". In: *Molecular cancer therapeutics* 15.6 (2016), pp. 1155–1162. DOI: 10.1158/1535-7163.mct-15-0843.

[141] Å. Flobak, B. Niederdorfer, V. T. Nakstad, L. Thommesen, G. Klinkenberg, and A. Lægreid. "A high-throughput drug combination screen of targeted small molecule inhibitors in cancer cell lines". In: *Scientific data* 6.1 (2019), p. 237. DOI: 10.1038/s41597-019-0255-7.

[142] Y. Lai, X. Wei, S. Lin, L. Qin, L. Cheng, and P. Li. "Current status and perspectives of patient-derived xenograft models in cancer research". In: *Journal of hematology & oncology* 10 (2017), pp. 1–14. DOI: 10.1186/s13045-017-0470-7.

[143] H. Fiebig. "Comparison of tumor response in nude mice and in patients". In: *Human tumour xenografts in anticancer drug development*. Springer, 1988, pp. 25–30. DOI: 10.1007/978-3-642-73252-2_4.

[144] E Izumchenko, K Paz, D Ciznadija, I Sloma, A Katz, D Vasquez-Dunddel, I Ben-Zvi, J Stebbing, W McGuire, W Harris, et al. "Patient-derived xenografts effectively capture responses to oncology therapy in a heterogeneous cohort of patients with solid tumors". In: *Annals of Oncology* 28.10 (2017), pp. 2595–2605. DOI: 10.1093/annonc/mdx416.

[145] I Fichtner, W Slisow, J Gill, M Becker, B Elbe, T Hillebrand, and M Bibby. "Anticancer drug response and expression of molecular markers in early-passage xenotransplanted colon carcinomas". In: *European journal of cancer* 40.2 (2004), pp. 298–307. DOI: 10.1016/j.ejca.2003.10.011.

[146] M. R. Kuracha, P. Thomas, B. W. Loggie, and V. Govindarajan. "Patient-derived xenograft mouse models of pseudomyxoma peritonei recapitulate the human inflammatory tumor microenvironment". In: *Cancer medicine* 5.4 (2016), pp. 711–719. DOI: 10.1002/cam4.640.

[147] D Guenot, E Guérin, S Aguillon-Romain, E Pencreach, A Schneider, A Neuville, M.-P. Chenard, I Duluc, S Du Manoir, C Brigand, et al. "Primary tumour genetic alterations and intra-tumoral heterogeneity are maintained in xenografts of human colon cancers showing chromosome instability". In: *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland* 208.5 (2006), pp. 643–652. DOI: 10.1002/path.1936.

[148] K. Kemper, O. Krijgsman, P. Cornelissen-Steijger, A. Shahrabi, F. Weeber, J.-Y. Song, T. Kuilman, D. J. Vis, L. F. Wessels, E. E. Voest, et al. "Intra-and inter-tumor heterogeneity in a vemurafenib-resistant melanoma patient and derived

xenografts”. In: *EMBO molecular medicine* 7.9 (2015), pp. 1104–1118. DOI: 10.15252/emmm.201404914.

[149] M. Bleijs, M. van de Wetering, H. Clevers, and J. Drost. “Xenograft and organoid model systems in cancer research”. In: *The EMBO journal* 38.15 (2019), e101654. DOI: 10.15252/embj.2019101654.

[150] Y. Liu, W. Wu, C. Cai, H. Zhang, H. Shen, and Y. Han. “Patient-derived xenograft models in cancer therapy: technologies and applications”. In: *Signal Transduction and Targeted Therapy* 8.1 (2023), p. 160. DOI: 10.1038/s41392-023-01419-2.

[151] H. Gao, J. M. Korn, S. Ferretti, J. E. Monahan, Y. Wang, M. Singh, C. Zhang, C. Schnell, G. Yang, Y. Zhang, et al. “High-throughput screening using patient-derived tumor xenografts to predict clinical trial drug response”. In: *Nature medicine* 21.11 (2015), pp. 1318–1325. DOI: 10.1038/nm.3954.

[152] L. Huang, A. Holtzinger, I. Jagan, M. BeGora, I. Lohse, N. Ngai, C. Nostro, R. Wang, L. B. Muthuswamy, H. C. Crawford, et al. “Ductal pancreatic cancer modeling and drug screening using human pluripotent stem cell–and patient-derived tumor organoids”. In: *Nature medicine* 21.11 (2015), pp. 1364–1371. DOI: 10.1038/nm.3973.

[153] N. Sachs, J. De Ligt, O. Kopper, E. Gogola, G. Bounova, F. Weeber, A. V. Balgobind, K. Wind, A. Gracanin, H. Begthel, et al. “A living biobank of breast cancer organoids captures disease heterogeneity”. In: *Cell* 172.1 (2018), pp. 373–386. DOI: 10.1016/j.cell.2017.11.010.

[154] H. H. Yan, H. C. Siu, S. Law, S. L. Ho, S. S. Yue, W. Y. Tsui, D. Chan, A. S. Chan, S. Ma, K. O. Lam, et al. “A comprehensive human gastric cancer organoid biobank captures tumor subtype heterogeneity and enables therapeutic screening”. In: *Cell stem cell* 23.6 (2018), pp. 882–897. DOI: 10.1016/j.stem.2018.09.016.

[155] G. Vlachogiannis, S. Hedayat, A. Vatsiou, Y. Jamin, J. Fernández-Mateos, K. Khan, A. Lampis, K. Eason, I. Huntingford, R. Burke, et al. “Patient-derived organoids model treatment response of metastatic gastrointestinal cancers”. In: *Science* 359.6378 (2018), pp. 920–926. DOI: 10.1126/science.aao2774.

[156] L. Huang, B. Bockorny, I. Paul, D. Akshinthala, P.-O. Frappart, O. Gandarilla, A. Bose, V. Sanchez-Gonzalez, E. E. Rouse, S. D. Lehoux, et al. “PDX-derived organoids model in vivo drug response and secrete biomarkers”. In: *JCI insight* 5.21 (2020). DOI: 10.1172/jci.insight.135544.

[157] H. Tiriac, P. Belleau, D. D. Engle, D. Plenker, A. Deschênes, T. D. Somerville, F. E. Froeling, R. A. Burkhart, R. E. Denroche, G.-H. Jang, et al. "Organoid profiling identifies common responders to chemotherapy in pancreatic cancer". In: *Cancer discovery* 8.9 (2018), pp. 1112–1129. DOI: 10.1158/2159-8290.CD-18-0349.

[158] S. Mourragui, M. Loog, M. A. Van De Wiel, M. J. Reinders, and L. F. Wessels. "PRECISE: a domain adaptation approach to transfer predictors of drug response from pre-clinical models to tumors". In: *Bioinformatics* 35.14 (2019), pp. i510–i519. DOI: 10.1093/bioinformatics/btz372.

[159] S. M. Mourragui, M. Loog, D. J. Vis, K. Moore, A. G. Manjon, M. A. van de Wiel, M. J. Reinders, and L. F. Wessels. "Predicting patient response with models trained on cell lines and patient-derived xenografts by nonlinear transfer learning". In: *Proceedings of the National Academy of Sciences* 118.49 (2021), e2106682118. DOI: 10.1073/pnas.2106682118.

[160] J. Kong, H. Lee, D. Kim, S. K. Han, D. Ha, K. Shin, and S. Kim. "Network-based machine learning in colorectal and bladder organoid models predicts anti-cancer drug efficacy in patients". In: *Nature communications* 11.1 (2020), p. 5485. DOI: 10.1038/s41467-020-19313-8.

[161] T. L. Riss, R. A. Moravec, A. L. Niles, S. Duellman, H. A. Benink, T. J. Worzella, and L. Minor. "Cell viability assays". In: *Assay guidance manual [Internet]* (2016).

[162] Molecular Probes, Inc. *SYTO® Red Fluorescent Nucleic Acid Stains*. 2001. URL: https://www.thermofisher.com/document-connect/document-connect.html?url=https://assets.thermofisher.com/TFS-Assets%2FLSG%2Fmanuals%2Fmp11340.pdf (visited on 05/11/2025).

[163] S. Anoopkumar-Dukie, J. Carey, T Conere, E O'sullivan, F. Van Pelt, and A Allshire. "Resazurin assay of radiation response in cultured cells". In: *The British journal of radiology* 78.934 (2005), pp. 945–947. DOI: 10.1259/bjr/54004230.

[164] Promega Corporation. *CellTiter-Glo ® Luminescent Cell Viability Assay*. 2015. URL: https://www.promega.de/products/cell-health-assays/cell-viability-and-cytotoxicity-assays/celltiter_glo-luminescent-cell-viability-assay/?catNum=G7570 (visited on 05/11/2025).

[165] H. Lightfoot, D. van der Meer, and D. J. Vis. *gdscIC50: Pipeline for GDSC Curve Fitting*. R package version 0.99.4. 2021.

[166] D. J. Vis, L. Bombardelli, H. Lightfoot, F. Iorio, M. J. Garnett, and L. F. Wessels. "Multilevel models improve precision and speed of IC50 estimates". In: *Pharmacogenomics* 17.7 (2016), pp. 691–700. DOI: 10.2217/pgs.16.15.

[167] M. J. Garnett, E. J. Edelman, S. J. Heidorn, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares, et al. "Systematic identification of genomic markers of drug sensitivity in cancer cells". In: *Nature* 483.7391 (2012), pp. 570–575. DOI: 10.1038/nature11005.

[168] M. Hafner, M. Niepel, M. Chung, and P. K. Sorger. "Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs". In: *Nature methods* 13.6 (2016), pp. 521–527. DOI: 10.1038/nmeth.3853.

[169] N. Pozdeyev, M. Yoo, R. Mackie, R. E. Schweppe, A. C. Tan, and B. R. Haugen. "Integrating heterogeneous drug sensitivity data from cancer pharmacogenomic studies". In: *Oncotarget* 7.32 (2016), p. 51619. DOI: 10.18632/oncotarget.10010.

[170] M. Hafner, M. Niepel, and P. K. Sorger. "Alternative drug sensitivity metrics improve preclinical cancer pharmacogenomics". In: *Nature biotechnology* 35.6 (2017), pp. 500–502. DOI: 10.1038/nbt.3882.

[171] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. *An introduction to statistical learning with applications in R*. Vol. Second Edition. Springer, 2021. DOI: 10.1007/978-1-0716-1418-1.

[172] Z. Safikhani, N. El-Hachem, R. Quevedo, P. Smirnov, A. Goldenberg, N. J. Birkbak, C. Mason, C. Hatzis, L. Shi, H. J. Aerts, et al. "Assessment of pharmacogenomic agreement". In: *F1000Research* 5 (2016). DOI: 10.12688/f1000research.8705.1.

[173] H. Sharifi-Noghabi, S. Jahangiri-Tazehkand, P. Smirnov, C. Hon, A. Mammoliti, S. K. Nair, A. S. Mer, M. Ester, and B. Haibe-Kains. "Drug sensitivity prediction from cell line-based pharmacogenomics data: guidelines for developing machine learning models". In: *Briefings in Bioinformatics* 22.6 (2021), bbab294. DOI: 10.1093/bib/bbab294.

[174] D. R. Liston and M. Davis. "Clinically Relevant Concentrations of Anticancer Drugs: A Guide for Nonclinical StudiesGuide to Clinical Exposures of Anticancer Drugs". In: *Clinical cancer research* 23.14 (2017), pp. 3489–3498. DOI: 10.1158/1078-0432.ccr-16-3083.

[175] B. Haibe-Kains, N. El-Hachem, N. J. Birkbak, A. C. Jin, A. H. Beck, H. J. Aerts, and J. Quackenbush. "Inconsistency in large pharmacogenomic studies". In: *Nature* 504.7480 (2013), pp. 389–393. DOI: 10.1038%2Fnature12831.

[176] J. C. Costello, L. M. Heiser, E. Georgii, M. Gönen, M. P. Menden, N. J. Wang, M. Bansal, M. Ammad-Ud-Din, P. Hintsanen, S. A. Khan, et al. "A community effort to assess and improve drug sensitivity prediction algorithms". In: *Nature biotechnology* 32.12 (2014), pp. 1202–1212. DOI: 10.1038/nbt.2877.

[177] Z. Stanfield, M. Coşkun, and M. Koyutürk. "Drug response prediction as a link prediction problem". In: *Scientific reports* 7.1 (2017), p. 40321. DOI: 10.1038/srep40321.

[178] Y. He, J. Liu, and X. Ning. "Drug selection via joint push and learning to rank". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17.1 (2018), pp. 110–123. DOI: 10.1109/tcbb.2018.2848908.

[179] R. Su, X. Liu, L. Wei, and Q. Zou. "Deep-Resp-Forest: a deep forest model to predict anti-cancer drug response". In: *Methods* 166 (2019), pp. 91–102. DOI: 10.1016/j.ymeth.2019.02.009.

[180] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik. *cluster: Cluster Analysis Basics and Extensions*. R package version 1.14.2 — For new features, see the 'Changelog' file (in the package source). 2021. URL: https://CRAN.R-project.org/package=cluster.

[181] L. Kaufman and P. J. Rousseeuw. "Finding Groups in Data: An Introduction to Cluster Analysis". In: Springer US, 2009. Chap. 2. ISBN: 9780387399409. DOI: 10.1002/9780470316801.ch2.

[182] L. Kaufman, P. Rousseeuw, F. of Mathematics, and I. (Delft). *Clustering by Means of Medoids*. Delft University of Technology : reports of the Faculty of Technical Mathematics and Informatics. Faculty of Mathematics and Informatics, 1987. URL: https://books.google.de/books?id=HK-4GwAACAAJ.

[183] A. Malyutina, M. M. Majumder, W. Wang, A. Pessia, C. A. Heckman, and J. Tang. "Drug combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer". In: *PLoS computational biology* 15.5 (2019), e1006752. DOI: 10.1371/journal.pcbi.1006752.

[184] B. Yadav, K. Wennerberg, T. Aittokallio, and J. Tang. "Searching for drug synergy in complex dose–response landscapes using an interaction potency model". In: *Computational and structural biotechnology journal* 13 (2015), pp. 504–513. DOI: 10.1016/j.csbj.2015.09.001.

[185] S Loewe. "The problem of synergism and antagonism of combined drugs". In: *Arzneimittel-forschung* 3.6 (1953), pp. 285–290.

[186] C. I. Bliss. "The toxicity of poisons applied jointly 1". In: *Annals of applied biology* 26.3 (1939), pp. 585–615. DOI: 10.1111/j.1744-7348.1939.tb06990.x.

[187] M. C. Berenbaum. "What is synergy?" In: *Pharmacological reviews* 41.2 (1989), pp. 93–141. DOI: 10.1016/S0031-6997(25)00026-2.

[188] A. Ianevski, A. K. Giri, and T. Aittokallio. “SynergyFinder 2.0: visual analytics of multi-drug combination synergies”. In: *Nucleic acids research* 48.W1 (2020), W488–W493. DOI: 10.1093/nar/gkaa216.

[189] *SynergyFinder - User Documentation.* Note: website not available anymore, archived version available at https://web.archive.org/web/20250115014914/https://synergyfinder.fimm.fi/synergy/synfin_docs/. URL: https://synergyfinder.fimm.fi/synergy/synfin_docs/ (visited on 01/15/2025).

[190] M. J. Evans and J. S. Rosenthal. “Probability and statistics: The science of uncertainty”. In: 2nd edition. WH Freeman, 2009. Chap. 1.3. ISBN: 9781429224628.

[191] Wellcome Sanger Institute, GDSC database. *News.* 2023. URL: https://www.cancerrxgene.org/news (visited on 05/11/2025).

[192] Wellcome Sanger Institute, GDSC database. *Resources Download - IC50 Data definitions.* 2024. URL: https://cog.sanger.ac.uk/cancerrxgene/GDSC_release8.5/GDSC_Fitted_Data_Description.pdf (visited on 11/22/2024).

[193] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. “Exploration, normalization, and summaries of high density oligonucleotide array probe level data”. In: *Biostatistics* 4.2 (2003), pp. 249–264. DOI: 10.1093/biostatistics/4.2.249.

[194] M. P. Menden, D. Wang, M. J. Mason, B. Szalai, K. C. Bulusu, Y. Guan, T. Yu, J. Kang, M. Jeon, R. Wolfinger, et al. “Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen”. In: *Nature communications* 10.1 (2019), p. 2674. DOI: 10.1038/s41467-019-09799-2.

[195] T. B. Brown. “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165* (2020). DOI: 10.48550/arXiv.2005.14165.

[196] OpenAI. *DALL-E: Creating images from text.* 2021. URL: https://openai.com/dall-e (visited on 05/11/2025).

[197] V. Kaul, S. Enslin, and S. A. Gross. “History of artificial intelligence in medicine”. In: *Gastrointestinal Endoscopy* 92.4 (2020), pp. 807–812. ISSN: 0016-5107. DOI: 10.1016/j.gie.2020.06.040.

[198] W. H. Walters and E. I. Wilder. “Fabrication and errors in the bibliographic citations generated by ChatGPT”. In: *Scientific Reports* 13.1 (2023), p. 14045. DOI: 10.1038/s41598-023-41032-5.

[199] M. Mathys, M. Willi, and R. Meier. “Synthetic Photography Detection: A Visual Guidance for Identifying Synthetic Images Created by AI”. In: *arXiv preprint arXiv:2408.06398* (2024). DOI: 10.48550/arXiv.2408.06398.

[200] T. Davenport and R. Kalakota. “The potential for artificial intelligence in healthcare”. In: *Future healthcare journal* 6.2 (2019), pp. 94–98. DOI: 10.7861%2Ffuturehosp.6-2-94.

[201] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. “Dissecting racial bias in an algorithm used to manage the health of populations”. In: *Science* 366.6464 (2019), pp. 447–453. DOI: 10.1126/science.aax2342.

[202] I. El Naqa and M. J. Murphy. “Machine Learning in Radiation Oncology: Theory and Applications”. In: Springer International Publishing, 2015. Chap. I.1.3. ISBN: 9783319183053. DOI: 10.1007/978-3-319-18305-3_1.

[203] O. Chapelle, B. Scholkopf, and A. Zien. “Semi-Supervised Learning”. In: MIT Press, 2006. Chap. 1.1. DOI: 10.7551/mitpress/9780262033589.001.0001.

[204] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 2.1. DOI: 10.1007/978-1-0716-1418-1.

[205] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 2.4. DOI: 10.1007/978-0-387-84858-7.

[206] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. In: https://www.deeplearningbook.org. MIT Press, 2016. Chap. 5.2.

[207] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 2.2. DOI: 10.1007/978-1-0716-1418-1.

[208] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274.

[209] K. Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2 (1991), pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T.

[210] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 14.1. DOI: 10.1007/978-0-387-84858-7.

[211] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006. DOI: 10.7551/mitpress/9780262033589.001.0001.

[212] J. E. Van Engelen and H. H. Hoos. “A survey on semi-supervised learning”. In: *Machine learning* 109.2 (2020), pp. 373–440. DOI: 10.1007/s10994-019-05855-6.

[213] L. Rampášek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg. “Dr.VAE: improving drug response prediction via modeling of drug perturbation effects”. In: *Bioinformatics* 35.19 (2019), pp. 3743–3751. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz158.

[214] L. P. Kaelbling, M. L. Littman, and A. W. Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285. DOI: 10.1613/jair.301.

[215] M. Liu, X. Shen, and W. Pan. “Deep reinforcement learning for personalized treatment recommendation”. In: *Statistics in medicine* 41.20 (2022), pp. 4034–4056. DOI: 10.1002/sim.9491.

[216] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 3.2. DOI: 10.1007/978-1-0716-1418-1.

[217] A. E. Hoerl and R. W. Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67. DOI: 10.2307/1267351.

[218] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[219] H. Zou and T. Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.

[220] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 3.4. DOI: 10.1007/978-0-387-84858-7.

[221] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 4.4. DOI: 10.1007/978-1-0716-1418-1.

[222] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 4. DOI: 10.1007/978-1-0716-1418-1.

[223] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. “An introduction to statistical learning with applications in R, Second Edition”. In: Springer, 2021. Chap. 9.1. DOI: 10.1007/978-1-0716-1418-1.

[224] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 12.2. DOI: 10.1007/978-0-387-84858-7.

[225] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 9.2. DOI: 10.1007/978-1-0716-1418-1.

[226] T. Hastie, J. Friedman, R. Tibshirani, et al. "The Elements of Statistical Learning, Second Edition". In: Springer series in statistics New York, 2009. Chap. 12.3. DOI: 10.1007/978-0-387-84858-7.

[227] J. C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. Ed. by B. Schölkopf, C. J. Burges, and V. Vapnik. MIT Press, 1999, pp. 61–74.

[228] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 9.4. DOI: 10.1007/978-1-0716-1418-1.

[229] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. "Support vector regression machines". In: *Advances in neural information processing systems* 9 (1996).

[230] E. Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*. Vol. 1. USAF school of Aviation Medicine, 1985. DOI: 10.2307/1403797.

[231] B. W. Silverman and M. C. Jones. "E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)". In: *International Statistical Review/Revue Internationale de Statistique* (1989), pp. 233–238. DOI: 10.2307/1403796.

[232] T. Cover and P. Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27. DOI: 10.1109/TIT.1967.1053964.

[233] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 3.5. DOI: 10.1007/978-1-0716-1418-1.

[234] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 8.2. DOI: 10.1007/978-1-0716-1418-1.

[235] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 8.2. DOI: 10.1007/978-1-0716-1418-1.

[236] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[237] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 15.4. DOI: 10.1007/978-0-387-84858-7.

[238] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 15.3. DOI: 10.1007/978-0-387-84858-7.

[239] Y. Freund and R. E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139. DOI: 10.1006/jcss.1997.1504.

[240] T. Hastie, S. Rosset, J. Zhu, and H. Zou. “Multi-class adaboost”. In: *Statistics and its Interface* 2.3 (2009), pp. 349–360. DOI: 10.4310/sii.2009.v2.n3.a8.

[241] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. https://www.deeplearningbook.org. MIT Press, 2016.

[242] C. De Niz, R. Rahman, X. Zhao, and R. Pal. “Algorithms for Drug Sensitivity Prediction”. In: 9.4 (2016). DOI: 10.3390/a9040077.

[243] J. Chen and L. Zhang. “A survey and systematic assessment of computational methods for drug response prediction”. In: *Briefings in bioinformatics* 22.1 (2021), pp. 232–246. DOI: 10.1093/bib/bbz164.

[244] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. In: https://www.deeplearningbook.org. MIT Press, 2016. Chap. 6.

[245] T. Hastie, J. Friedman, R. Tibshirani, et al. “The Elements of Statistical Learning, Second Edition”. In: Springer series in statistics New York, 2009. Chap. 11.3. DOI: 10.1007/978-0-387-84858-7.

[246] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. In: https://www.deeplearningbook.org. MIT Press, 2016. Chap. 6.3.

[247] P. Oosting. “Signal transmission in the nervous system”. In: *Reports on Progress in Physics* 42.9 (1979), p. 1479. DOI: 10.1088/0034-4885/42/9/001.

[248] K. He, X. Zhang, S. Ren, and J. Sun. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the International Conference on Computer Vision*. 2015, pp. 1026–1034. DOI: 10.1109/iccv.2015.123.

[249] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 2010, pp. 249–256.

[250] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. In: https://www.deeplearningbook.org. MIT Press, 2016. Chap. 8.

[251] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536. DOI: 10.7551/mitpress/1888.003.0013.

[252] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 10.2. DOI: 10.1007/978-1-0716-1418-1.

[253] T. Hastie, J. Friedman, R. Tibshirani, et al. "The Elements of Statistical Learning, Second Edition". In: Springer series in statistics New York, 2009. Chap. 11.4. DOI: 10.1007/978-0-387-84858-7.

[254] European Commission. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS*. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206 (visited on 05/11/2025).

[255] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha. "Secure and robust machine learning for healthcare: A survey". In: *IEEE Reviews in Biomedical Engineering* 14 (2020), pp. 156–180. DOI: 10.1109/RBME.2020.3013489.

[256] European Commission. *Requirements of Trustworthy AI*. 2019. URL: https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines/1.html (visited on 05/11/2025).

[257] M. Naser and A. H. Alavi. "Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences". In: *Architecture, Structures and Construction* 3.4 (2023), pp. 499–517. DOI: 10.1007/s44150-021-00015-8.

[258] C. Spearman. "The proof and measurement of association between two things." In: (1961). DOI: 10.1037/11491-005.

[259] M. G. Kendall. "A NEW MEASURE OF RANK CORRELATION". In: *Biometrika* 30.1-2 (June 1938), pp. 81–93. ISSN: 0006-3444. DOI: 10.1093/biomet/30.1-2.81.

[260] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. "The Balanced Accuracy and Its Posterior Distribution". In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 3121–3124. DOI: 10.1109/ICPR.2010.764.

[261] B. W. Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451. DOI: 10.1016/0005-2795(75)90109-9.

[262] D. J. Hand. “Measuring classifier performance: a coherent alternative to the area under the ROC curve”. In: *Machine Learning* 77.1 (June 2009), 103–123. ISSN: 1573-0565. DOI: 10.1007/s10994-009-5119-5. URL: http://dx.doi.org/10.1007/s10994-009-5119-5.

[263] T. Fawcett. “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (June 2006), 861–874. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2005.10.010.

[264] B. Kompa, J. Snoek, and A. L. Beam. “Second opinion needed: communicating uncertainty in medical machine learning”. In: *NPJ Digital Medicine* 4.1 (2021), p. 4. DOI: 10.1038/s41746-020-00367-3.

[265] C. Gruber, P. O. Schenk, M. Schierholz, F. Kreuter, and G. Kauermann. “Sources of Uncertainty in Machine Learning–A Statisticians’ View”. In: *arXiv preprint arXiv:2305.16703* (2023). DOI: 10.48550/arXiv.2305.16703.

[266] E. Hüllermeier and W. Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine Learning* 110 (2021), pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

[267] A. N. Angelopoulos and S. Bates. “A gentle introduction to conformal prediction and distribution-free uncertainty quantification”. In: *arXiv preprint arXiv:2107.07511* (2021). DOI: 10.1561/9781638281597.

[268] L. Serafino. *On the de Finetti’s representation theorem: an evergreen (and often misunderstood) result at the foundation of Statistics*. 2016. URL: https://philsci-archive.pitt.edu/12012/ (visited on 05/11/2025).

[269] V. Vovk. “Conditional validity of inductive conformal predictors”. In: *Asian conference on machine learning*. PMLR. 2012, pp. 475–490. DOI: 10.1007/s10994-013-5355-6.

[270] A. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan. “Uncertainty sets for image classifiers using conformal prediction”. In: *arXiv preprint arXiv:2009.14193* (2020). DOI: 10.48550/arXiv.2009.14193.

[271] Y. Romano, M. Sesia, and E. Candes. “Classification with valid and adaptive coverage”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3581–3591. DOI: 10.48550/arXiv.2006.02544.

[272] U. Norinder, L. Carlsson, S. Boyer, and M. Eklund. “Introducing conformal prediction in predictive modeling. A transparent and flexible alternative to applicability domain determination”. In: *Journal of chemical information and modeling* 54.6 (2014), pp. 1596–1603. DOI: 10.1021/ci5001168.

[273] Y. Romano, E. Patterson, and E. Candes. "Conformalized quantile regression". In: *Advances in neural information processing systems* 32 (2019). DOI: 10.48550/arXiv.1905.03222.

[274] A. Partin, T. S. Brettin, Y. Zhu, O. Narykov, A. Clyde, J. Overbeek, and R. L. Stevens. "Deep learning methods for drug response prediction in cancer: predominant and emerging trends". In: *Frontiers in medicine* 10 (2023), p. 1086097. DOI: 10.3389/fmed.2023.1086097.

[275] Z. C. Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57. DOI: 10.1145/3236386.3241340.

[276] Y. Lou, R. Caruana, and J. Gehrke. "Intelligible models for classification and regression". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 150–158. DOI: 10.1145/2339530.2339556.

[277] L. S. Shapley et al. "A value for n-person games". In: *Contribution to the Theory of Games* 2 (1953). DOI: 10.1515/9781400881970-018.

[278] S. M. Lundberg and S.-I. Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017). DOI: 10.48550/arXiv.1705.07874.

[279] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint* (2013). arXiv:1312.6034. DOI: 10.48550/arXiv.1312.6034.

[280] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)". In: *arXiv preprint arXiv:1711.11279* (2017). DOI: 10.48550/arXiv.1711.11279.

[281] J. Crabbé and M. van der Schaar. "Concept activation regions: A generalized framework for concept-based explanations". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 2590–2607. DOI: 10.48550/arXiv.2209.11222.

[282] M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester, and J. Saez-Rodriguez. "Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties". In: *PLoS one* 8.4 (2013), e61318. DOI: 10.1371/journal.pone.0061318.

[283] N. Zhang, H. Wang, Y. Fang, J. Wang, X. Zheng, and X. S. Liu. "Predicting anticancer drug responses using a dual-layer integrated cell line-drug network model". In: *PLoS computational biology* 11.9 (2015), e1004498. DOI: 10.1371/journal.pcbi.1004498.

[284] L. Wang, X. Li, L. Zhang, and Q. Gao. "Improved anticancer drug response prediction in cell lines using matrix factorization with similarity regularization". In: *BMC Cancer* 17.1 (2017), pp. 1–12. DOI: 10.1186/s12885-017-3500-5.

[285] R. Rahman, K. Matlock, S. Ghosh, and R. Pal. "Heterogeneity aware random forest for drug sensitivity prediction". In: *Scientific reports* 7.1 (2017), pp. 1–11. DOI: 10.1038/s41598-017-11665-4.

[286] K. Matlock, C. De Niz, R. Rahman, S. Ghosh, and R. Pal. "Investigation of model stacking for drug sensitivity prediction". In: *BMC bioinformatics* 19.3 (2018), pp. 21–33. DOI: 10.1186/s12859-018-2060-2.

[287] J. D. Janizek, S. Celik, and S.-I. Lee. "Explainable machine learning prediction of synergistic drug combinations for precision cancer medicine". In: *BioRxiv* (2018), p. 331769. DOI: 10.1101/331769.

[288] X. He, L. Folkman, and K. Borgwardt. "Kernelized rank learning for personalized drug recommendation". In: *Bioinformatics* 34.16 (2018), pp. 2808–2816. DOI: 10.1093/bioinformatics/bty132.

[289] Y. Chang, H. Park, H.-J. Yang, S. Lee, K.-Y. Lee, T. S. Kim, J. Jung, and J.-M. Shin. "Cancer drug response profile scan (CDRscan): a deep learning model that predicts drug effectiveness from cancer genomic signature". In: *Scientific reports* 8.1 (2018), p. 8857. DOI: 10.1038/s41598-018-27214-6.

[290] Y. Fang, P. Xu, J. Yang, and Y. Qin. "A quantile regression forest based method to predict drug response and assess prediction reliability". In: *PLoS One* 13.10 (2018), e0205155. DOI: 10.1371/journal.pone.0205155.

[291] H. Liu, Y. Zhao, L. Zhang, and X. Chen. "Anti-cancer drug response prediction using neighbor-based collaborative filtering with global effect removal". In: *Molecular Therapy-Nucleic Acids* 13 (2018), pp. 303–311. DOI: 10.1016/j.omtn.2018.09.011.

[292] Y.-C. Chiu, H.-I. H. Chen, T. Zhang, S. Zhang, A. Gorthi, L.-J. Wang, Y. Huang, and Y. Chen. "Predicting drug response of tumors from integrated genomic profiles by deep neural networks". In: *BMC medical genomics* 12.1 (2019), pp. 143–155. DOI: 10.1186/s12920-018-0460-9.

[293] A. Oskooei, M. Manica, R. Mathis, and M. R. Martínez. “Network-based biased tree ensembles (NetBiTE) for drug sensitivity prediction and drug sensitivity biomarker identification in cancer”. In: *Scientific reports* 9.1 (2019), p. 15918. DOI: 10.1038/s41598-019-52093-w.

[294] L. Deng, Y. Cai, W. Zhang, W. Yang, B. Gao, and H. Liu. “Pathway-guided deep neural network toward interpretable and predictive modeling of drug sensitivity”. In: *Journal of Chemical Information and Modeling* 60.10 (2020), pp. 4497–4505. DOI: 10.1021/acs.jcim.0c00331.

[295] K. T. Ahmed, S. Park, Q. Jiang, Y. Yeu, T. Hwang, and W. Zhang. “Network-based drug sensitivity prediction”. In: *BMC medical genomics* 13.11 (2020), pp. 1–10. DOI: 10.1186/s12920-020-00829-3.

[296] F. Ahmadi Moughari and C. Eslahchi. “ADRML: anticancer drug response prediction using manifold learning”. In: *Scientific reports* 10.1 (2020), p. 14245. DOI: 10.1038/s41598-020-71257-7.

[297] Y.-C. Tang and A. Gottlieb. “Explainable drug sensitivity prediction through cancer pathway enrichment”. In: *Scientific reports* 11.1 (2021), pp. 1–10. DOI: 10.1038/s41598-021-82612-7.

[298] O. Bazgir, S. Ghosh, and R. Pal. “Investigation of REFINED CNN ensemble learning for anti-cancer drug sensitivity prediction”. In: *Bioinformatics* 37.Supplement 1 (July 2021), pp. i42–i50. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab336.

[299] T. Nguyen, G. T. Nguyen, T. Nguyen, and D.-H. Le. “Graph convolutional networks for drug response prediction”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 19.1 (2021), pp. 146–154. DOI: doi.org/10.1109/TCBB.2021.3060430.

[300] S. Chawla, A. Rockstroh, M. Lehman, E. Ratther, A. Jain, A. Anand, A. Gupta, N. Bhattacharya, S. Poonia, P. Rai, et al. “Gene expression based inference of cancer drug sensitivity”. In: *Nature communications* 13.1 (2022), p. 5680. DOI: 10.1038/s41467-022-33291-z.

[301] M. Gönen and A. A. Margolin. “Drug susceptibility prediction against a panel of drugs using kernelized Bayesian multitask learning”. In: *Bioinformatics* 30.17 (2014), pp. i556–i563. DOI: 10.1093/bioinformatics/btu464.

[302] K. Preuer, R. P. Lewis, S. Hochreiter, A. Bender, K. C. Bulusu, and G. Klambauer. “DeepSynergy: predicting anti-cancer drug synergy with Deep Learning”. In: *Bioinformatics* 34.9 (2018), pp. 1538–1546. DOI: 10.1093/bioinformatics/btx806.

[303] Q. Liu, Z. Hu, R. Jiang, and M. Zhou. "DeepCDR: a hybrid graph convolutional network for predicting cancer drug response". In: *Bioinformatics* 36.Supplement_2 (2020), pp. i911–i918. DOI: 10.1093/bioinformatics/btaa822.

[304] Y. Kim, S. Zheng, J. Tang, W. Jim Zheng, Z. Li, and X. Jiang. "Anticancer drug synergy prediction in understudied tissues using transfer learning". In: *Journal of the American Medical Informatics Association* 28.1 (2021), pp. 42–51. DOI: 10.1093/jamia/ocaa212.

[305] H. I. Kuru, O. Tastan, and A. E. Cicek. "MatchMaker: a deep learning framework for drug synergy prediction". In: *IEEE/ACM transactions on computational biology and bioinformatics* 19.4 (2021), pp. 2334–2344. DOI: 10.1109/tcbb.2021.3086702.

[306] X. Li, Y. Xu, H. Cui, T. Huang, D. Wang, B. Lian, W. Li, G. Qin, L. Chen, and L. Xie. "Prediction of synergistic anti-cancer drug combinations based on drug target network and drug induced gene expression profiles". In: *Artificial intelligence in medicine* 83 (2017), pp. 35–43. DOI: 10.1016/j.artmed.2017.05.008.

[307] F. Zhang, M. Wang, J. Xi, J. Yang, and A. Li. "A novel heterogeneous network-based method for drug response prediction in cancer cell lines". In: *Scientific reports* 8.1 (2018), p. 3355. DOI: 10.1038/s41598-018-21622-4.

[308] O. Chapelle, B. Scholkopf, and A. Zien. "Semi-Supervised Learning". In: MIT Press, 2006. Chap. 1.2. DOI: 10.7551/mitpress/9780262033589.001.0001.

[309] W. Jia, M. Sun, J. Lian, and S. Hou. "Feature dimensionality reduction: a review". In: *Complex & Intelligent Systems* 8.3 (2022), pp. 2663–2693. DOI: 10.1007/s40747-021-00637-x.

[310] T. Hastie, J. Friedman, R. Tibshirani, et al. "The Elements of Statistical Learning, Second Edition". In: Springer series in statistics New York, 2009. Chap. 18.3. DOI: 10.1007/978-0-387-84858-7.

[311] I. Goodfellow, Y. Bengio, and A. Courville. "Deep Learning". In: https://www.deeplearningbook.org. MIT Press, 2016. Chap. 14.

[312] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. "An introduction to statistical learning with applications in R, Second Edition". In: Springer, 2021. Chap. 6.3. DOI: 10.1007/978-1-0716-1418-1.

[313] F. Codicè, C. Pancotti, C. Rollo, Y. Moreau, P. Fariselli, and D. Raimondi. "The specification game: rethinking the evaluation of drug response prediction for precision oncology". In: *Journal of Cheminformatics* 17.1 (2025), p. 33. DOI: 10.1186/s13321-025-00972-y.

[314] H. Sharifi-Noghabi, O. Zolotareva, C. C. Collins, and M. Ester. “MOLI: multi-omics late integration with deep neural networks for drug response prediction”. In: *Bioinformatics* 35.14 (2019), pp. i501–i509. DOI: 10.1093/bioinformatics/btz318.

[315] K. Lee, D. Cho, J. Jang, K. Choi, H.-o. Jeong, J. Seo, W.-K. Jeong, and S. Lee. “RAMP: response-aware multi-task learning with contrastive regularization for cancer drug response prediction”. In: *Briefings in Bioinformatics* 24.1 (2023), bbac504. DOI: 10.1093/bib/bbac504.

[316] L. Grinsztajn, E. Oyallon, and G. Varoquaux. “Why do tree-based models still outperform deep learning on typical tabular data?” In: *Advances in neural information processing systems* 35 (2022), pp. 507–520. DOI: 10.48550/arXiv.2207.08815.

[317] R. Shwartz-Ziv and A. Armon. “Tabular data: Deep learning is not all you need”. In: *Information Fusion* 81 (2022), pp. 84–90. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2021.11.011.

[318] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci. “Deep Neural Networks and Tabular Data: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.6 (2024), pp. 7499–7519. DOI: 10.1109/tnnls.2022.3229161.

[319] S. Kapoor and A. Narayanan. “Leakage and the reproducibility crisis in machine-learning-based science”. In: *Patterns* 4.9 (2023). DOI: 10.1016/j.patter.2023.100804.

[320] K. Koras, D. Juraeva, J. Kreis, J. Mazur, E. Staub, and E. Szczurek. “Feature selection strategies for drug sensitivity prediction”. In: *Scientific Reports* 10.1 (2020), p. 9377. DOI: 10.1038/s41598-020-65927-9.

[321] X. Liu, C. Song, F. Huang, H. Fu, W. Xiao, and W. Zhang. “GraphCDR: a graph neural network method with contrastive learning for cancer drug response prediction”. In: *Briefings in Bioinformatics* 23.1 (2022), bbab457. DOI: 10.1093/bib/bbab457.

[322] X. Cheng, C. Dai, Y. Wen, X. Wang, X. Bo, S. He, and S. Peng. “NeRD: a multichannel neural network to predict cellular response of drugs by integrating multidimensional data”. In: *BMC medicine* 20.1 (2022), p. 368. DOI: 10.1186/s12916-022-02549-0.

[323] H. Wang, C. Dai, Y. Wen, X. Wang, W. Liu, S. He, X. Bo, and S. Peng. “GADRP: graph convolutional networks and autoencoders for cancer drug response prediction”. In: *Briefings in Bioinformatics* 24.1 (2023), bbac501. DOI: 10.1093/bib/bbac501.

[324] P. Jia, R. Hu, and Z. Zhao. “Benchmark of embedding-based methods for accurate and transferable prediction of drug response”. In: *Briefings in Bioinformatics* 24.3 (2023), bbad098. DOI: 10.1093/bib/bbad098.

[325] H. Liu, Y. Zhao, L. Zhang, and X. Chen. “Anti-cancer Drug Response Prediction Using Neighbor-Based Collaborative Filtering with Global Effect Removal”. In: *Molecular Therapy - Nucleic Acids* 13 (2018), pp. 303–311. ISSN: 2162-2531. DOI: 10.1016/j.omtn.2018.09.011.

[326] R. Rahman and R. Pal. “Analyzing drug sensitivity prediction based on dose response curve characteristics”. In: *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. 2016, pp. 140–143. DOI: 10.1109/bhi.2016.7455854.

[327] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. “Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent”. In: *Journal of Statistical Software* 39.5 (2011). R package version 4.1.1, pp. 1–13. DOI: 10.18637/jss.v039.i05.

[328] M. N. Wright and A. Ziegler. “ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R”. In: *Journal of Statistical Software* 77.1 (2017). R package version 0.12.1, pp. 1–17. DOI: 10.18637/jss.v077.i01.

[329] B. Greenwell, B. Boehmke, J. Cunningham, et al. *gbm: Generalized Boosted Regression Models*. R package version 2.1.8. 2020. URL: https://CRAN.R-project.org/package=gbm (visited on 05/11/2025).

[330] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL: https://www.tensorflow.org/ (visited on 05/11/2025).

[331] F. Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras (visited on 05/11/2025).

[332] F. Martínez-Jiménez, F. Muiños, I. Sentís, J. Deu-Pons, I. Reyes-Salazar, C. Arnedo-Pac, L. Mularoni, O. Pich, J. Bonet, H. Kranas, et al. “A compendium of mutational cancer driver genes”. In: *Nature Reviews Cancer* 20.10 (2020), pp. 555–572. DOI: 10.1038/s41568-020-0290-x.

[333] W. Kirch, ed. *Encyclopedia of Public Health: Volume 1: A-H*. Springer Dordrecht, 2008. DOI: 10.1007/978-1-4020-5614-7.

[334] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, et al. “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles”. In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545 –15550. DOI: 10.1073/pnas.0506580102.

[335] A. N. Kolmogorov. “Sulla determinazione empirica di una legge didistribuzione”. In: *Giornale dell’Instituto Italiano degli Attuari* 4 (1933), pp. 83–91.

[336] N. V. Smirnov. “Estimate of deviation between empirical distribution functions in two independent samples”. In: *Bulletin Moscow University* 2.2 (1939), pp. 3–16.

[337] Y. Benjamini and Y. Hochberg. “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), pp. 289 –300. DOI: 10.1111/j.2517-6161.1995.tb02031.x.

[338] C. Ding and H. Peng. “Minimum redundancy feature selection from microarray gene expression data”. In: *Journal of bioinformatics and computational biology* 3.02 (2005), pp. 185–205. DOI: 10.1142/S0219720005001004.

[339] N. Kwak and C.-H. Choi. “Input feature selection for classification problems”. In: *IEEE transactions on neural networks* 13.1 (2002), pp. 143–159. DOI: 10.1109/72.977291.

[340] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[341] I. Karagiannaki, Y. Pantazis, E. Chatzaki, and I. Tsamardinos. “Pathway Activity Score Learning for Dimensionality Reduction of Gene Expression Data”. In: *Discovery Science*. Springer International Publishing, 2020, pp. 246–261. DOI: 10.1007/978-3-030-61527-7_17.

[342] M. Kanehisa and S. Goto. “KEGG: kyoto encyclopedia of genes and genomes”. In: *Nucleic acids research* 28.1 (2000), pp. 27–30. DOI: 10.1093/nar/gkac963.

[343] M. Gillespie, B. Jassal, R. Stephan, M. Milacic, K. Rothfels, A. Senff-Ribeiro, J. Griss, C. Sevilla, L. Matthews, C. Gong, et al. “The reactome pathway knowledgebase 2022”. In: *Nucleic acids research* 50.D1 (2022), pp. D687–D692. DOI: 10.1093/nar/gkab1028.

[344] D. Nishimura. “BioCarta”. In: *Biotech Software & Internet Report* 2.3 (2001), pp. 117–120. DOI: 10.1089/152791601750294344.

[345] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: https://www.R-project.org/ (visited on 05/11/2025).

[346] J. Muschelli. *matlabr: An Interface for MATLAB using System Calls*. R package version 1.6.0. 2020.

[347] N. Gerstner et al. *GeneTrail 3*. 2021. URL: https://github.com/unisb-bioinf/genetrail3 (visited on 05/11/2025).

[348] N. Gerstner, T. Kehl, K. Lenhof, A. Müller, C. Mayer, L. Eckhart, N. L. Grammes, C. Diener, M. Hart, O. Hahn, J. Walter, T. Wyss-Coray, E. Meese, A. Keller, and H.-P. Lenhof. "GeneTrail 3: advanced high-throughput enrichment analysis". In: *Nucleic Acids Research* 48.W1 (2020), W515–W520. ISSN: 0305-1048. DOI: 10.1093/nar/gkaa306.

[349] A. J. Minn, C. M. Rudin, L. H. Boise, and C. B. Thompson. "Expression of bcl-xL can confer a multidrug resistance phenotype". In: *Blood* 86.5 (1995), pp. 1903–10. DOI: 10.1182/blood.v86.5.1903.bloodjournal8651903.

[350] A. Zaanan, K. Okamoto, H. Kawakami, K. Khazaie, S. Huang, and F. A. Sinicrope. "The mutant KRAS gene up-regulates BCL-XL protein via STAT3 to confer apoptosis resistance that is reversed by BIM protein induction and BCL-XL antagonism". In: *Journal of Biological Chemistry* 290.39 (2015), pp. 23838–23849. DOI: 10.1074/jbc.m115.657833.

[351] Y.-L. Lo and Y. Liu. "Reversing Multidrug Resistance in Caco-2 by Silencing MDR1, MRP1, MRP2, and BCL-2/BCL-xL Using Liposomal Antisense Oligonucleotides". In: *PLOS ONE* 9.3 (2014), pp. 1–14. DOI: 10.1371/journal.pone.0090180. URL: 10.1371/journal.pone.0090180.

[352] H. Park, S.-Y. Cho, H. Kim, D. Na, J. Y. Han, J. Chae, C. Park, O.-K. Park, S. Min, J. Kang, et al. "Genomic alterations in ¡i¿BCL2L1¡/i¿ and ¡i¿DLC1¡/i¿ contribute to drug sensitivity in gastric cancer". In: *Proceedings of the National Academy of Sciences* 112.40 (2015), pp. 12492–12497. DOI: 10.1073/pnas.1507491112.

[353] Q. Gao, G. Zhang, Y. Zheng, Y. Yang, C. Chen, J. Xia, L. Liang, C. Lei, Y. Hu, X. Cai, et al. "SLC27A5 deficiency activates NRF2/TXNRD1 pathway by increased lipid peroxidation in HCC". In: *Cell Death & Differentiation* 27 (2020), pp. 1086–1104. DOI: 10.1038/s41418-019-0399-1.

[354] M. Delgobo, R. M. Gonçalves, M. A. Delazeri, M. Falchetti, A. Zandoná, R. N. das Neves, K. Almeida, A. C. Fagundes, D. P. Gelain, J. I. Fracasso, et al. "Thioredoxin reductase-1 levels are associated with NRF2 pathway activation and tumor recurrence in non-small cell lung cancer". In: *Free Radical Biology and Medicine* 177 (2021), pp. 58–71. DOI: 10.1016/j.freeradbiomed.2021.10.020.

[355] X.-J. Wang, Z. Sun, N. F. Villeneuve, S. Zhang, F. Zhao, Y. Li, W. Chen, X. Yi, W. Zheng, G. T. Wondrak, et al. "Nrf2 enhances resistance of cancer cells to chemotherapeutic drugs, the dark side of Nrf2". In: *Carcinogenesis* 29.6 (2008), pp. 1235–1243. DOI: 10.1093/carcin/bgn095.

[356] X. An, X. Chen, D. Yi, H. Li, and Y. Guan. “Representation of molecules for drug response prediction”. In: *Briefings in Bioinformatics* 23.1 (2022), bbab393. DOI: 10.1093/bib/bbab393.

[357] H. Zhang, Y. Chen, and F. Li. “Predicting anticancer drug response with deep learning constrained by signaling pathways”. In: *Frontiers in Bioinformatics* 1 (2021), p. 639349. DOI: 10.3389/fbinf.2021.639349.

[358] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. “On the class imbalance problem”. In: *2008 Fourth international conference on natural computation*. Vol. 4. IEEE. 2008, pp. 192–201. DOI: 10.1109/ICNC.2008.871.

[359] R. P. Ribeiro and N. Moniz. “Imbalanced regression and extreme value prediction”. In: *Machine Learning* 109.9 (2020), pp. 1803–1835. DOI: 10.1007/s10994-020-05900-9.

[360] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[361] A. Liaw and M. Wiener. “Classification and Regression by randomForest”. In: *R News* 2.3 (2002), pp. 18–22. URL: https://CRAN.R-project.org/doc/Rnews/.

[362] *RandomForestRegressor*. URL: https://www.scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html (visited on 05/11/2025).

[363] L. Pasqualini, H. Bu, M. Puhr, N. Narisu, J. Rainer, B. Schlick, G. Schäfer, M. Angelova, Z. Trajanoski, S. T. Börno, M. R. Schweiger, C. Fuchsberger, and H. Klocker. “miR-22 and miR-29a Are Members of the Androgen Receptor Cistrome Modulating LAMC1 and Mcl-1 in Prostate Cancer”. In: *Molecular Endocrinology* 29.7 (2015), pp. 1037–1054. DOI: 10.1210/me.2014-1358.

[364] L. Zhang, C. Li, and X. Su. “Emerging impact of the long noncoding RNA MIR22HG on proliferation and apoptosis in multiple human cancers”. In: *Journal of Experimental & Clinical Cancer Research* 39.1 (2020). DOI: 10.1186/s13046-020-01784-8.

[365] S. M. Chan, D. Thomas, M. R. Corces-Zimmerman, S. Xavy, S. Rastogi, W.-J. Hong, F. Zhao, B. C. Medeiros, D. A. Tyvoll, and R. Majeti. “Isocitrate dehydrogenase 1 and 2 mutations induce BCL-2 dependence in acute myeloid leukemia”. In: *Nature Medicine* 21.2 (2015), pp. 178–184. DOI: 10.1038/nm.3788.

[366] M. Rahmani, M. M. Aust, E. Hawkins, R. E. Parker, M. Ross, M. Kmieciak, L. B. Reshko, K. A. Rizzo, C. I. Dumur, A. Ferreira-Gonzalez, and S. Grant. "Co-administration of the mTORC1/TORC2 inhibitor INK128 and the Bcl-2/Bcl-xL antagonist ABT-737 kills human myeloid leukemia cells through Mcl-1 down-regulation and AKT inactivation". In: *Haematologica* 100.12 (2015), pp. 1553–1563. DOI: 10.3324/haematol.2015.130351.

[367] I. R. Logan, H. V. McNeill, S. Cook, X. Lu, J. Lunec, and C. N. Robson. "Analysis of the MDM2 antagonist nutlin-3 in human prostate cancer cells". In: *The Prostate* 67.8 (2007), pp. 900–906. DOI: 10.1002/pros.20568.

[368] K. I. Pishas, S. J. Neuhaus, M. T. Clayer, A. W. Schreiber, D. M. Lawrence, M. Perugini, R. J. Whitfield, G. Farshid, J. Manavis, S. Chryssidis, B. J. Mayo, R. C. Haycox, K. Ho, M. P. Brown, R. J. D'Andrea, A. Evdokiou, D. M. Thomas, J. Desai, D. F. Callen, and P. M. Neilsen. "Nutlin-3a Efficacy in Sarcoma Predicted by Transcriptomic and Epigenetic Profiling". In: *Cancer Research* 74.3 (2013), pp. 921–931. DOI: 10.1158/0008-5472.can-13-2424.

[369] M. Zanjirband, R. J. Edmondson, and J. Lunec. "Pre-clinical efficacy and synergistic potential of the MDM2-p53 antagonists, Nutlin-3 and RG7388, as single agents and in combined treatment with cisplatin in ovarian cancer". In: *Oncotarget* 7.26 (2016), pp. 40115–40134. DOI: 10.18632/oncotarget.9499.

[370] K. Kumamoto, E. A. Spillare, K. Fujita, I. Horikawa, T. Yamashita, E. Appella, M. Nagashima, S. Takenoshita, J. Yokota, and C. C. Harris. "Nutlin-3a Activates p53 to Both Down-regulate Inhibitor of Growth 2 and Up-regulate mir-34a, mir-34b, and mir-34c Expression, and Induce Senescence". In: *Cancer Research* 68.9 (2008), pp. 3193–3203. DOI: 10.1158/0008-5472.can-07-2780.

[371] M. Kumawat, R. Singh, I. Karuna, N. Ahlawat, and S. Ahlawat. "Salmonella Typhimurium peptidyl-prolyl cis–trans isomerase C (PPIase C) plays a substantial role in protein folding to maintain the protein structure". In: *World Journal of Microbiology and Biotechnology* 36.11 (2020), p. 168. DOI: 10.1007/s11274-020-02943-x.

[372] A. Elfenbein and M. Simons. "Syndecan-4 signaling at a glance". In: *Journal of cell science* 126.17 (2013), pp. 3799–3804. DOI: 10.1242/jcs.124636.

[373] P. Xie, F.-Q. Yuan, M.-S. Huang, W. Zhang, H.-H. Zhou, X. Li, and Z.-Q. Liu. "DCBLD2 affects the development of colorectal cancer via EMT and angiogenesis and modulates 5-FU drug resistance". In: *Frontiers in Cell and Developmental Biology* 9 (2021), p. 669285. DOI: 10.3389/fcell.2021.669285.

[374] S. Cheng, L.-Y. Wang, C.-H. Wang, F.-K. Wang, B. Zhu, P. Zhang, and G.-H. Wang. "Transmembrane protein DCBLD2 is correlated with poor prognosis and affects phenotype by regulating epithelial-mesenchymal transition in human glioblastoma cells". In: *Neuroreport* 32.6 (2021), pp. 507–517. DOI: 10.1097/wnr.0000000000001611.

[375] J. He, H. Huang, Y. Du, D. Peng, Y. Zhou, Y. Li, H. Wang, Y. Zhou, and Y. Nie. "Association of DCBLD2 upregulation with tumor progression and poor survival in colorectal cancer". In: *Cellular Oncology* 43 (2020), pp. 409–420. DOI: 10.1007/s13402-020-00495-8.

[376] L. Schneider, T. Kehl, K. Thedinga, N. L. Grammes, C. Backes, C. Mohr, B. Schubert, K. Lenhof, N. Gerstner, A. D. Hartkopf, et al. "ClinOmicsTrailbc: a visual analytics tool for breast cancer treatment stratification". In: *Bioinformatics* 35.24 (2019), pp. 5171–5181. DOI: 10.1093/bioinformatics/btz302.

[377] J. Alvarsson, S. A. McShane, U. Norinder, and O. Spjuth. "Predicting with confidence: using conformal prediction in drug discovery". In: *Journal of Pharmaceutical Sciences* 110.1 (2021), pp. 42–49. DOI: 10.1016/j.xphs.2020.09.055.

[378] A. Morger, M. Mathea, J. H. Achenbach, A. Wolf, R. Buesen, K.-J. Schleifer, R. Landsiedel, and A. Volkamer. "KnowTox: pipeline and case study for confident prediction of potential toxic effects of compounds in early phases of development". In: *Journal of Cheminformatics* 12.1 (2020), p. 24. DOI: 10.1186/s13321-020-00422-x.

[379] A. Morger, F. Svensson, S. Arvidsson McShane, N. Gauraha, U. Norinder, O. Spjuth, and A. Volkamer. "Assessing the calibration in toxicological in vitro models with conformal prediction". In: *Journal of Cheminformatics* 13.1 (2021), p. 35. DOI: 10.1186/s13321-021-00511-5.

[380] N. Meinshausen and G. Ridgeway. "Quantile regression forests." In: *Journal of machine learning research* 7.6 (2006).

[381] F. Yassaee Meybodi and C. Eslahchi. "Predicting anti-cancer drug response by finding optimal subset of drugs". In: *Bioinformatics* 37.23 (2021), pp. 4509–4516. DOI: 10.1093/bioinformatics/btab466.

[382] Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *arXiv preprint arXiv:1506.02142* (2015). DOI: 10.48550/arXiv.1506.02142.

[383] B. Efron. "Jackknife-after-bootstrap standard errors and influence functions". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 54.1 (1992), pp. 83–111. DOI: 10.1111/j.2517-6161.1992.tb01866.x.

[384] Z. Dong, N. Zhang, C. Li, H. Wang, Y. Fang, J. Wang, and X. Zheng. "Anticancer drug sensitivity prediction in cell lines from baseline gene expression through recursive feature selection". In: *BMC cancer* 15.1 (2015), p. 489. DOI: 10.1186/s12885-015-1492-6.

[385] J. Vazquez and J. C. Facelli. "Conformal prediction in clinical medical sciences". In: *Journal of Healthcare Informatics Research* 6.3 (2022), pp. 241–252. DOI: 10.1007/s41666-021-00113-8.

[386] A. Torkamannia, Y. Omidi, and R. Ferdousi. "A review of machine learning approaches for drug synergy prediction in cancer". In: *Briefings in Bioinformatics* 23.3 (2022), bbac075. ISSN: 1477-4054. DOI: 10.1093/bib/bbac075. eprint: https://academic.oup.com/bib/article-pdf/23/3/bbac075/43745080/bbac075.pdf.

[387] R. Rahman and R. Pal. "Analyzing drug sensitivity prediction based on dose response curve characteristics". In: *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. 2016, pp. 140–143. DOI: 10.1109/bhi.2016.7455854.

[388] R. Rahman, S. R. Dhruba, S. Ghosh, and R. Pal. "Functional random forest with applications in dose-response predictions". In: *Scientific reports* 9.1 (2019), p. 1628. DOI: 10.1038/s41598-018-38231-w.

[389] H. Julkunen, A. Cichonska, P. Gautam, S. Szedmak, J. Douat, T. Pahikkala, T. Aittokallio, and J. Rousu. "Leveraging multi-way interactions for systematic prediction of pre-clinical drug combination effects". In: *Nature communications* 11.1 (2020), p. 6136. DOI: 10.1038/s41467-020-19950-z.

[390] A. Ianevski, A. K. Giri, P. Gautam, A. Kononov, S. Potdar, J. Saarela, K. Wennerberg, and T. Aittokallio. "Prediction of drug combination effects with a minimal set of experiments". In: *Nature machine intelligence* 1.12 (2019), pp. 568–577. DOI: 10.1038/s42256-019-0122-4.

[391] A. C. Palmer and P. K. Sorger. "Combination cancer therapy can confer benefit via patient-to-patient variability without drug additivity or synergy". In: *Cell* 171.7 (2017), pp. 1678–1691. DOI: 10.1016/j.cell.2017.11.009.

[392] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. "Reoptimization of MDL keys for use in drug discovery". In: *Journal of chemical information and computer sciences* 42.6 (2002), pp. 1273–1280. DOI: 10.1021/ci010132r.

[393] G. Landrum et al. *RDKit Documentation - rdkit.Chem.Descriptors module.* URL: https://www.rdkit.org/docs/source/rdkit.Chem.Descriptors.html (visited on 05/11/2025).

[394] G. Landrum et al. *RDKit: Open-source cheminformatics.* version 2023.3.2. DOI: 10.5281/zenodo.8053810. URL: http://www.rdkit.org (visited on 05/11/2025).

[395] D. Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36. DOI: 10.1021/ci00057a005.

[396] C. Ritz, F. Baty, J. C. Streibig, and D. Gerhard. "Dose-Response Analysis Using R". In: *PLOS ONE* 10.e0146021 (12 2015). DOI: 10.1371/journal.pone.0146021.

[397] *drc: Analysis of Dose-Response Curves.* URL: https://cran.r-project.org/web/packages/drc/drc.pdf (visited on 05/11/2025).

[398] H. W. Borchers. *pracma: Practical Numerical Math Functions.* R package version 2.4.2. 2022. URL: https://CRAN.R-project.org/package=pracma (visited on 05/11/2025).

[399] P. Liu, H. Li, S. Li, and K.-S. Leung. "Improving prediction of phenotypic drug response on cancer cell lines using deep convolutional network". In: *BMC Bioinformatics* 20.408 (2019). ISSN: 1471-2105. DOI: 10.1186/s12859-019-2910-6.

[400] T. Pivetta, F. Isaia, F. Trudu, A. Pani, M. Manca, D. Perra, F. Amato, and J. Havel. "Development and validation of a general approach to predict and quantify the synergism of anti-cancer drugs using experimental design and artificial neural networks". In: *Talanta* 115 (2013), pp. 84–93. DOI: 10.1016/j.talanta.2013.04.031.

[401] J. Gu, X. Zhang, Y. Ma, N. Li, F. Luo, L. Cao, Z. Wang, G. Yuan, L. Chen, W. Xiao, et al. "Quantitative modeling of dose–response and drug combination based on pathway network". In: *Journal of cheminformatics* 7 (2015), pp. 1–10. DOI: 10.1186/s13321-015-0066-6.

[402] J. Yang, H Tang, Y Li, R Zhong, T Wang, S. Wong, G Xiao, and Y Xie. "DIGRE: Drug-Induced Genomic Residual Effect Model for Successful Prediction of Multidrug Effects". In: *CPT: pharmacometrics & systems pharmacology* 4.2 (2015), pp. 91–97. DOI: 10.1002/psp4.1.

[403] A. Zimmer, I. Katzir, E. Dekel, A. E. Mayo, and U. Alon. "Prediction of multidimensional drug dose responses based on measurements of drug pairs". In: *Proceedings of the National Academy of Sciences* 113.37 (2016), pp. 10442–10447. DOI: 10.1073/pnas.1606301113.

[404] Y.-C. Hsu, Y.-C. Chiu, Y. Chen, T.-H. Hsiao, and E. Y. Chuang. "A simple gene set-based method accurately predicts the synergy of drug pairs". In: *BMC Systems Biology* 10 (2016), pp. 313–322. DOI: 10.1186/s12918-016-0310-3.

[405] A. Zimmer, A. Tendler, I. Katzir, A. Mayo, and U. Alon. “Prediction of drug cocktail effects when the number of measurements is limited”. In: *PLoS biology* 15.10 (2017), e2002518. DOI: 10.1371/journal.pbio.2002518.

[406] M. Jeon, S. Kim, S. Park, H. Lee, and J. Kang. “In silico drug combination discovery for personalized cancer therapy”. In: *BMC systems biology* 12 (2018), pp. 59–67. DOI: 10.1186/s12918-018-0546-1.

[407] H. Li, S. Hu, N. Neamati, and Y. Guan. “TAIJI: approaching experimental replicates-level accuracy for drug synergy prediction”. In: *Bioinformatics* 35.13 (2019), pp. 2338–2339. DOI: 10.1093/bioinformatics/bty955.

[408] M. B. M. A. Rashid, T. B. Toh, L. Hooi, A. Silva, Y. Zhang, P. F. Tan, A. L. Teh, N. Karnani, S. Jha, C.-M. Ho, et al. “Optimizing drug combinations against multiple myeloma using a quadratic phenotypic optimization platform (QPOP)”. In: *Science translational medicine* 10.453 (2018), eaan0941. DOI: 10.1126/scitranslmed.aan0941.

[409] F. Xia, M. Shukla, T. Brettin, C. Garcia-Cardona, J. Cohn, J. E. Allen, S. Maslov, S. L. Holbeck, J. H. Doroshow, Y. A. Evrard, et al. “Predicting tumor cell line response to drug pairs with deep learning”. In: *BMC bioinformatics* 19 (2018), pp. 71–79. DOI: 10.1186/s12859-018-2509-3.

[410] P. Sidorov, S. Naulaerts, J. Ariey-Bonnet, E. Pasquier, and P. J. Ballester. “Predicting synergism of cancer drug combinations using NCI-ALMANAC data”. In: *Frontiers in chemistry* 7 (2019), p. 509. DOI: 10.3389/fchem.2019.00509.

[411] A. Ling and R. S. Huang. “Computationally predicting clinical drug combination efficacy with cancer cell line screens and independent drug action”. In: *Nature communications* 11.1 (2020), p. 5848. DOI: 10.1038/s41467-020-19563-6.

[412] C. Correia, A. Ferreira, J. Santos, R. Lapa, M. Yliperttula, A. Urtti, and N. Vale. “New in vitro-in silico approach for the prediction of in vivo performance of drug combinations”. In: *Molecules* 26.14 (2021), p. 4257. DOI: 10.3390/molecules26144257.

[413] P. Pinoli, G. Ceddia, S. Ceri, and M. Masseroli. “Predicting drug synergism by means of non-negative matrix tri-factorization”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 19.4 (2021), pp. 1956–1967. DOI: 10.1109/tcbb.2021.3091814.

[414] Cancer Cell Line Encyclopedia Consortium and Genomics of Drug Sensitivity in Cancer Consortium and others. “Pharmacogenomic agreement between two cancer cell line data sets”. In: *Nature* 528.7580 (2015), pp. 84–87. DOI: 10.1038/nature15736.

[415] D. Rogers and M. Hahn. "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5 (2010), pp. 742–754. DOI: 10.1021/ci100050t.

[416] D. Jiang, Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu, and T. Hou. "Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models". In: *Journal of cheminformatics* 13 (2021), pp. 1–23. DOI: 10.1186/s13321-020-00479-8.

[417] H. Liu, W. Zhang, B. Zou, J. Wang, Y. Deng, and L. Deng. "DrugCombDB: a comprehensive database of drug combinations toward the discovery of combinatorial therapy". In: *Nucleic acids research* 48.D1 (2020), pp. D871–D881. DOI: 10.1093/nar/gkz1007.

[418] C. Hatzis, P. L. Bedard, N. J. Birkbak, A. H. Beck, H. J. Aerts, D. F. Stern, L. Shi, R. Clarke, J. Quackenbush, and B. Haibe-Kains. "Enhancing reproducibility in cancer drug screening: how do we move forward?" In: *Cancer research* 74.15 (2014), pp. 4016–4023. DOI: 10.1158/0008-5472.CAN-14-0725.

[419] H. Sharifi-Noghabi, S. Peng, O. Zolotareva, C. C. Collins, and M. Ester. "AITL: adversarial inductive transfer learning with input and output space adaptation for pharmacogenomics". In: *Bioinformatics* 36.Supplement_1 (2020), pp. i380–i388. DOI: 10.1093/bioinformatics/btaa442.

[420] R. L. Siegel, K. D. Miller, N. S. Wagle, and A. Jemal. "Cancer statistics, 2023". In: *CA: a cancer journal for clinicians* 73.1 (2023), pp. 17–48. DOI: 10.3322/caac.21763.

[421] K. Saginala, A. Barsouk, J. S. Aluru, P. Rawla, S. A. Padala, and A. Barsouk. "Epidemiology of bladder cancer". In: *Medical sciences* 8.1 (2020), p. 15. DOI: 10.3390/medsci8010015.

[422] D. S. Kaufman, W. U. Shipley, and A. S. Feldman. "Bladder cancer". In: *The Lancet* 374.9685 (2009), pp. 239–249. DOI: 10.1016/S0140-6736(09)60491-8.

[423] National Cancer Institute (NCI), Surveillance, Epidemiology, and End Results Program (SEER). *Urinary Bladder (Invasive & In Situ) SEER 5-Year Relative Survival Rates, 2014-2020.* URL: https://seer.cancer.gov/statistics-network/explorer/application.html?site=71&data_type=4&graph_type=2&compareBy=race&chk_race_1=1&relative_survival_interval=5&sex=1&age_range=1&stage=101&advopt_precision=1&advopt_show_ci=on&hdn_view=0&advopt_show_apc=on&advopt_display=2#resultsRegion0 (visited on 05/11/2025).

[424] J. A. Witjes, H. M. Bruins, A. Carrión, R. Cathomas, E. Compérat, J. A. Efstathiou, R. Fietkau, G. Gakis, A. Lorch, A. Martini, et al. "European association of urology guidelines on muscle-invasive and metastatic bladder cancer: summary of the 2023 guidelines". In: *European urology* 85.1 (2024), pp. 17–31. DOI: 10.1016/j.eururo.2023.08.016.

[425] E. E. F. van de Putte, W. Otto, A. Hartmann, S. Bertz, R. Mayr, J. Bründl, J. Breyer, Q. Manach, E. M. Compérat, J. L. Boormans, et al. "Metric substage according to micro and extensive lamina propria invasion improves prognostics in T1 bladder cancer". In: *Urologic Oncology: Seminars and Original Investigations.* Vol. 36. 8. Elsevier. 2018, 361–e7. DOI: 10.1016/j.urolonc.2018.05.007.

[426] L. Dyrskjøt, T. Reinert, F. Algaba, E. Christensen, D. Nieboer, G. G. Hermann, K. Mogensen, W. Beukers, M. Marquez, U. Segersten, et al. "Prognostic impact of a 12-gene progression score in non–muscle-invasive bladder cancer: a prospective multicentre validation study". In: *European Urology* 72.3 (2017), pp. 461–469. DOI: 10.1016/j.eururo.2017.05.040.

[427] I. Vannini, F. Fanini, and M. Fabbri. "Emerging roles of microRNAs in cancer". In: *Current opinion in genetics & development* 48 (2018), pp. 128–133. DOI: 10.1016/j.gde.2018.01.001.

[428] S. Swarbrick, N. Wragg, S. Ghosh, and A. Stolzing. "Systematic review of miRNA as biomarkers in Alzheimer's disease". In: *Molecular neurobiology* 56 (2019), pp. 6156–6167. DOI: 10.1007/s12035-019-1500-y.

[429] S. Baumgart, P. Meschkat, P. Edelmann, J. Heinzelmann, A. Pryalukhin, R. Bohle, J. Heinzelbecker, M. Stöckle, and K. Junker. "MicroRNAs in tumor samples and urinary extracellular vesicles as a putative diagnostic tool for muscle-invasive bladder cancer". In: *Journal of Cancer Research and Clinical Oncology* 145 (2019), pp. 2725–2736. DOI: 10.1007/s00432-019-03035-6.

[430] C. Corrado, S. Raimondo, A. Chiesi, F. Ciccia, G. De Leo, and R. Alessandro. "Exosomes as intercellular signaling organelles involved in health and disease: basic science and clinical applications". In: *International journal of molecular sciences* 14.3 (2013), pp. 5338–5366. DOI: 10.3390/ijms14035338.

[431] U. E. Gibson, C. A. Heid, and P. M. Williams. "A novel method for real time quantitative RT-PCR." In: *Genome research* 6.10 (1996), pp. 995–1001. DOI: 10.1101/gr.6.10.995.

[432] D. Rodríguez-Lázaro and M. Hernández. "Real-time PCR in food science: introduction". In: *Current issues in molecular biology* 15.2 (2013), pp. 25–38. DOI: 10.21775/cimb.015.025.

[433] C. Zhan, L. Yan, L. Wang, W. Jiang, Y. Zhang, J. Xi, L. Chen, Y. Jin, Y. Qiao, Y. Shi, et al. “Identification of reference miRNAs in human tumors by TCGA miRNA-seq data”. In: *Biochemical and Biophysical Research Communications* 453.3 (2014), pp. 375–378. DOI: 10.1016/j.bbrc.2014.09.086.

[434] F. Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987. DOI: 10.2307/3001968.

[435] H. B. Mann and D. R. Whitney. “On a test of whether one of two random variables is stochastically larger than the other”. In: *The annals of mathematical statistics* (1947), pp. 50–60. DOI: 10.1214/aoms/1177730491.

[436] J. H. Zar. “Biostatistical analysis”. In: Pearson Education India, 1999. Chap. 8.11.

[437] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller. “pROC: an open-source package for R and S+ to analyze and compare ROC curves”. In: *BMC Bioinformatics* 12 (2011), p. 77. DOI: 10.1186/1471-2105-12-77.

[438] M. Culp, K. Johnson, and G. Michailidis. *ada: The R Package Ada for Stochastic Boosting*. R package version 2.0-5. 2016. URL: https://CRAN.R-project.org/package=ada.

[439] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. “kernlab – An S4 Package for Kernel Methods in R”. In: *Journal of Statistical Software* 11.9 (2004), pp. 1–20. DOI: 10.18637/jss.v011.i09.

[440] GraphPad Software. *GraphPad Prism*. URL: www.graphpad.com (visited on 05/11/2025).

[441] Y. Xue, L. Tong, F. LiuAnwei Liu, A. Liu, S. Zeng, Q. Xiong, Z. Yang, X. He, Y. Sun, and C. Xu. “Tumor-infiltrating M2 macrophages driven by specific genomic alterations are associated with prognosis in bladder cancer”. In: *Oncology reports* 42.2 (2019), pp. 581–594. DOI: 10.3892/or.2019.7196.

[442] R. Dai, Q. Jiang, Y. Zhou, R. Lin, H. Lin, Y. Zhang, J. Zhang, and X. Gao. “Lnc-STYK1-2 regulates bladder cancer cell proliferation, migration, and invasion by targeting miR-146b-5p expression and AKT/STAT3/NF-kB signaling”. In: *Cancer cell international* 21 (2021), pp. 1–14. DOI: 10.1186/s12935-021-02114-4.

[443] N. Mehterov, A. Sacconi, C. Pulito, B. Vladimirov, G. Haralanov, D. Pazardjikliev, B. Nonchev, I. Berindan-Neagoe, G. Blandino, and V. Sarafian. “A novel panel of clinically relevant miRNAs signature accurately differentiates oral cancer from normal mucosa”. In: *Frontiers in oncology* 12 (2022), p. 1072579. DOI: 10.3389/fonc.2022.1072579.

[444] Z. Qiu, H. Li, J. Wang, and C. Sun. “miR-146a and miR-146b in the diagnosis and prognosis of papillary thyroid carcinoma”. In: *Oncology Reports* 38.5 (2017), pp. 2735–2740. DOI: 10.3892/or.2017.5994.

[445] Z. Zhang, X. Fu, Y. Gao, Z. Nie, et al. “LINC01535 Attenuates ccRCC progression through regulation of the miR-146b-5p/TRIM2 axis and inactivation of the PI3K/Akt pathway”. In: *Journal of Oncology* 2022 (2022). DOI: 10.1155/2022/2153337.

[446] L. Shi, Y. Su, Z. Zheng, J. Qi, W. Wang, and C. Wang. “miR-146b-5p promotes colorectal cancer progression by targeting TRAF6”. In: *Experimental and Therapeutic Medicine* 23.3 (2022), pp. 1–12. DOI: 10.3892/etm.2022.11155.

[447] Y. Ren, X. Wang, T. Ji, and X. Cai. “MicroRNA-146b-5p suppresses cholangiocarcinoma cells by targeting TRAF6 and modulating p53 translocation”. In: *Acta histochemica* 123.7 (2021), p. 151793. DOI: 10.1016/j.acthis.2021.151793.

[448] B. Ouyang, N. Pan, H. Zhang, C. Xing, and W. Ji. “miR-146b-5p inhibits tumorigenesis and metastasis of gallbladder cancer by targeting Toll-like receptor 4 via the nuclear factor-κB pathway”. In: *Oncology Reports* 45.4 (2021). DOI: 10.3892/or.2021.7966.

[449] H. Wang and C.-P. Men. “Correlation of increased expression of MicroRNA-155 in bladder cancer and prognosis”. In: *Laboratory medicine* 46.2 (2015), pp. 118–122. DOI: 10.1309/lmwr9cea2k2xvsox.

[450] J. J. Lu, W. M. Yang, F. Li, W. Zhu, and Z. Chen. “Tunneling nanotubes mediated microRNA-155 intercellular transportation promotes bladder cancer cells’ invasive and proliferative capacity”. In: *International journal of nanomedicine* (2019), pp. 9731–9743. DOI: 10.2147/ijn.s217277.

[451] A. Awadalla, H. Abol-Enein, E. T. Hamam, A. E. Ahmed, S. M. Khirallah, A. El-Assmy, S. A. Mostafa, A. O. Babalghith, M. Ali, M. Abdel-Rahim, et al. “Identification of epigenetic interactions between miRNA and gene expression as potential prognostic markers in bladder cancer”. In: *Genes* 13.9 (2022), p. 1629. DOI: 10.3390/genes13091629.

[452] M. Li, J. Li, C. Ye, W. Wu, and Y. Cheng. “miR-200a-3p predicts prognosis and inhibits bladder cancer cell proliferation by targeting STAT4”. In: *Archives of Medical Science: AMS* 19.3 (2023), p. 724. DOI: 10.5114/aoms.2019.89969.

[453] S. Tan, Y. Kang, H. Li, H.-Q. He, L. Zheng, S.-Q. Wu, K. Ai, L. Zhang, R. Xu, X.-Z. Zhang, et al. “circST6GALNAC6 suppresses bladder cancer metastasis by sponging miR-200a-3p to modulate the STMN1/EMT axis”. In: *Cell death & disease* 12.2 (2021), p. 168. DOI: 10.1038/s41419-021-03459-4.

[454] R. Elango, K. A. Alsaleh, R. Vishnubalaji, M. Manikandan, A. M. Ali, N. Abd El-Aziz, A. Altheyab, A. Al-Rikabi, M. Alfayez, A. Aldahmash, et al. "MicroRNA expression profiling on paired primary and lymph node metastatic breast cancer revealed distinct microRNA profile associated with LNM". In: *Frontiers in oncology* 10 (2020), p. 756. DOI: 10.3389/fonc.2020.00756.

[455] Z. Di, M. Di, W. Fu, Q. Tang, Y. Liu, P. Lei, X. Gu, T. Liu, and M. Sun. "Integrated analysis identifies a nine-microRNA signature biomarker for diagnosis and prognosis in colorectal cancer". In: *Frontiers in Genetics* 11 (2020), p. 192. DOI: 10.3389/fgene.2020.00192.

[456] P. Wan, Z. Chen, M. Huang, H. Jiang, H. Wu, K. Zhong, G. Ding, and B. Wang. "miR-200a-3p facilitates bladder cancer cell proliferation by targeting the A20 gene". In: *Translational Andrology and Urology* 10.11 (2021), p. 4262. DOI: 10.21037/tau-21-941.

[457] I. Cavallari, A. Grassi, P. Del Bianco, A. Aceti, C. Zaborra, E. Sharova, I. Bertazzolo, D. M. D'Agostino, M. Iafrate, and V. Ciminale. "Prognostic stratification of bladder cancer patients with a microRNA-based approach". In: *Cancers* 12.11 (2020), p. 3133. DOI: 10.3390/cancers12113133.

[458] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76. DOI: 10.1109/JPROC.2020.3004555.

[459] J. Vanschoren. "Meta-learning". In: *Automated machine learning: methods, systems, challenges* (2019), pp. 35–61.

[460] R. Vilalta and Y. Drissi. "A perspective view and survey of meta-learning". In: *Artificial intelligence review* 18 (2002), pp. 77–95. DOI: 10.1023/A:1019956318069.

[461] S. M. Foltz, C. S. Greene, and J. N. Taroni. "Cross-platform normalization enables machine learning model training on microarray and RNA-seq data simultaneously". In: *Communications Biology* 6.1 (2023), p. 222. DOI: 10.1038/s42003-023-04588-6.

[462] L. Wu. *Mixed effects models for complex data*. Chapman and Hall/CRC, 2009. ISBN: 9780429142512.

[463] D. Maeser, W. Zhang, Y. Huang, and R. S. Huang. "A review of computational methods for predicting cancer drug response at the single-cell level through integration with bulk RNAseq data". In: *Current Opinion in Structural Biology* 84 (2024), p. 102745. DOI: 10.1016/j.sbi.2023.102745.

[464] B. Güvenç Paltun, H. Mamitsuka, and S. Kaski. "Improving drug response prediction by integrating multiple data sources: matrix factorization, kernel and network-based approaches". In: *Briefings in bioinformatics* 22.1 (2021), pp. 346–359. DOI: 10.1093/bib/bbz153.

[465] C. Wang, X. Lye, R. Kaalia, P. Kumar, and J. C. Rajapakse. "Deep learning and multi-omics approach to predict drug responses in cancer". In: *BMC bioinformatics* 22.Suppl 10 (2021), p. 632. DOI: 10.1186/s12859-022-04964-9.

[466] A. Park, Y. Lee, and S. Nam. "A performance evaluation of drug response prediction models for individual drugs". In: *Scientific Reports* 13.1 (2023), p. 11911. DOI: 10.1038/s41598-023-39179-2.

[467] M. Ali, S. A. Khan, K. Wennerberg, and T. Aittokallio. "Global proteomics profiling improves drug sensitivity prediction: results from a multi-omics, pan-cancer modeling approach". In: *Bioinformatics* 34.8 (2018), pp. 1353–1362. DOI: 10.1093/bioinformatics/btx766.

[468] D. Chakravarty, J. Gao, S. Phillips, R. Kundra, H. Zhang, J. Wang, J. E. Rudolph, R. Yaeger, T. Soumerai, M. H. Nissan, et al. "OncoKB: a precision oncology knowledge base". In: *JCO precision oncology* 1 (2017), pp. 1–16. DOI: 10.1200/PO.17.00011.

[469] M. Griffith, N. C. Spies, K. Krysiak, J. F. McMichael, A. C. Coffman, A. M. Danos, B. J. Ainscough, C. A. Ramirez, D. T. Rieke, L. Kujan, et al. "CIViC is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer". In: *Nature genetics* 49.2 (2017), pp. 170–174. DOI: 10.1038/ng.3774.

[470] S. Haider, R. Rahman, S. Ghosh, and R. Pal. "A copula based approach for design of multivariate random forests for drug sensitivity prediction". In: *PloS one* 10.12 (2015), e0144490. DOI: 10.1371/journal.pone.0144490.

[471] D. Wei, C. Liu, X. Zheng, and Y. Li. "Comprehensive anticancer drug response prediction based on a simple cell line-drug complex network model". In: *BMC bioinformatics* 20 (2019), p. 44. DOI: 10.1186/s12859-019-2608-9.

[472] C. Suphavilai, D. Bertrand, and N. Nagarajan. "Predicting cancer drug response using a recommender system". In: *Bioinformatics* 34.22 (2018), pp. 3907–3914. DOI: 10.1093/bioinformatics/bty452.

[473] A. Partin, T. Brettin, Y. A. Evrard, Y. Zhu, H. Yoo, F. Xia, S. Jiang, A. Clyde, M. Shukla, M. Fonstein, et al. "Learning curves for drug response prediction in cancer cell lines". In: *BMC bioinformatics* 22 (2021), pp. 1–18. DOI: 10.1186/s12859-021-04163-y.

[474] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang. "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742. DOI: 10.1109/TETCI.2021.3100641.

[475] M. Haddouchi and A. Berrado. "A survey and taxonomy of methods interpreting random forest models". In: *arXiv preprint arXiv:2407.12759* (2024). DOI: 10.48550/arXiv.2407.12759.

[476] J. Yang, K. Zhou, Y. Li, and Z. Liu. "Generalized out-of-distribution detection: a survey". In: *arXiv preprint arXiv:2110.11334* (2021). DOI: 10.48550/arXiv.2110.11334.

[477] J. Liu, Z. Shen, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui. "Towards out-of-distribution generalization: A survey". In: *arXiv preprint arXiv:2108.13624* (2021). DOI: 10.48550/arXiv.2108.13624.

[478] U.S. Food and Drug Administration. *Digital Health Innovation Action Plan.* 2017. URL: https://www.fda.gov/media/106331/download (visited on 05/11/2025).

[479] Y. Luo, L. Yin, W. Bai, and K. Mao. "An appraisal of incremental learning methods". In: *Entropy* 22.11 (2020), p. 1190. DOI: 10.3390/e22111190.

[480] D. Lyell, F. Magrabi, M. Z. Raban, L. G. Pont, M. T. Baysari, R. O. Day, and E. Coiera. "Automation bias in electronic prescribing". In: *BMC medical informatics and decision making* 17 (2017), pp. 1–10. DOI: 10.1186/s12911-017-0425-5.

[481] European Commission. *Proposal for a Directive of the European Parliament and of the Council on adapting non-contractual civil liability rules to artificial intelligence (AI Liability Directive) (AILD).* 2022. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52022PC0496 (visited on 05/11/2025).

[482] J. Lee. "Is artificial intelligence better than human clinicians in predicting patient outcomes?" In: *Journal of Medical Internet Research* 22.8 (2020), e19918. DOI: 10.2196/19918.

[483] M. H. Jarrahi. "Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making". In: *Business horizons* 61.4 (2018), pp. 577–586. DOI: 10.1016/j.bushor.2018.03.007.

[484] M. Ghandi, F. W. Huang, J. Jané-Valbuena, G. V. Kryukov, C. C. l. Lo, E. R. McDonald, J. Barretina, E. T. Gelfand, C. M. Bielski, H. Li, et al. "Next-generation characterization of the cancer cell line encyclopedia". In: *Nature* 569.7757 (2019), pp. 503–508. DOI: 10.1038/s41586-019-1186-3.

[485] M. Safran, I. Dalah, J. Alexander, N. Rosen, T. Iny Stein, M. Shmoish, N. Nativ, I. Bahir, T. Doniger, H. Krug, et al. "GeneCards Version 3: the human gene integrator". In: *Database* 2010 (2010). DOI: 10.1093/database/baq020.

[486] T. G. Whitsett, I. T. Mathews, M. H. Cardone, R. J. Lena, W. E. Pierceall, M. Bittner, C. Sima, J. LoBello, G. J. Weiss, and N. L. Tran. "Mcl-1 mediates TWEAK/Fn14-induced non–small cell lung cancer survival and therapeutic response". In: *Molecular Cancer Research* 12.4 (2014), pp. 550–559. DOI: 10.1158/1541-7786.mcr-13-0458.

[487] L. Zhang, C. Li, and X. Su. "Emerging impact of the long noncoding RNA MIR22HG on proliferation and apoptosis in multiple human cancers". In: *Journal of Experimental & Clinical Cancer Research* 39.1 (2020). DOI: 10.1186/s13046-020-01784-8.

[488] L. Pasqualini, H. Bu, M. Puhr, N. Narisu, J. Rainer, B. Schlick, G. Schäfer, M. Angelova, Z. Trajanoski, S. T. Börno, M. R. Schweiger, C. Fuchsberger, and H. Klocker. "miR-22 and miR-29a Are Members of the Androgen Receptor Cistrome Modulating LAMC1 and Mcl-1 in Prostate Cancer". In: *Molecular Endocrinology* 29.7 (2015), pp. 1037–1054. DOI: 10.1210/me.2014-1358.

[489] M. G. Rees, B. Seashore-Ludlow, J. H. Cheah, D. J. Adams, E. V. Price, S. Gill, S. Javaid, M. E. Coletti, V. L. Jones, N. E. Bodycombe, et al. "Correlating chemical sensitivity and basal gene expression reveals mechanism of action". In: *Nature chemical biology* 12.2 (2016), pp. 109–116. DOI: 10.1038/nchembio.1986.

[490] S. M. Chan, D. Thomas, M. R. Corces-Zimmerman, S. Xavy, S. Rastogi, W.-J. Hong, F. Zhao, B. C. Medeiros, D. A. Tyvoll, and R. Majeti. "Isocitrate dehydrogenase 1 and 2 mutations induce BCL-2 dependence in acute myeloid leukemia". In: *Nature Medicine* 21.2 (2015), pp. 178–184. DOI: 10.1038/nm.3788.

[491] M. Rahmani, M. M. Aust, E. Hawkins, R. E. Parker, M. Ross, M. Kmieciak, L. B. Reshko, K. A. Rizzo, C. I. Dumur, A. Ferreira-Gonzalez, and S. Grant. "Coadministration of the mTORC1/TORC2 inhibitor INK128 and the Bcl-2/Bcl-xL antagonist ABT-737 kills human myeloid leukemia cells through Mcl-1 down-regulation and AKT inactivation". In: *Haematologica* 100.12 (2015), pp. 1553–1563. DOI: 10.3324/haematol.2015.130351.

[492] K. T. Flaherty. "Moving Forward: Making BRAF-Targeted Therapy Better". In: *BRAF Targets in Melanoma: Biological Mechanisms, Resistance, and Drug Discovery* (2015), pp. 183–201. DOI: 10.1007/978-1-4939-2143-0_9.

[493] J. Eriksson, V. Le Joncour, P. Nummela, T. Jahkola, S. Virolainen, P. Laakkonen, O. Saksela, and E. Hölttä. “Gene expression analyses of primary melanomas reveal CTHRC1 as an important player in melanoma progression”. In: *Oncotarget* 7.12 (2016), p. 15065. DOI: 10.18632/oncotarget.7604.

[494] J. Li, J. Feng, Y Wang, X. Li, X. Chen, Y Su, Y. Shen, Y Chen, B Xiong, C. Yang, et al. “The B-RafV600E inhibitor dabrafenib selectively inhibits RIP3 and alleviates acetaminophen-induced liver injury”. In: *Cell death & disease* 5.6 (2014), e1278–e1278. DOI: 10.1038/cddis.2014.241.

[495] K. I. Pishas, S. J. Neuhaus, M. T. Clayer, A. W. Schreiber, D. M. Lawrence, M. Perugini, R. J. Whitfield, G. Farshid, J. Manavis, S. Chryssidis, B. J. Mayo, R. C. Haycox, K. Ho, M. P. Brown, R. J. D'Andrea, A. Evdokiou, D. M. Thomas, J. Desai, D. F. Callen, and P. M. Neilsen. “Nutlin-3a Efficacy in Sarcoma Predicted by Transcriptomic and Epigenetic Profiling”. In: *Cancer Research* 74.3 (2013), pp. 921–931. DOI: 10.1158/0008-5472.can-13-2424.

[496] T. Stoyanova, N. Roy, D. Kopanja, S. Bagchi, and P. Raychaudhuri. “DDB2 decides cell fate following DNA damage”. In: *Proceedings of the National Academy of Sciences* 106.26 (2009), pp. 10690–10695. DOI: 10.1073/pnas.0812254106.

[497] M. Zanjirband, R. J. Edmondson, and J. Lunec. “Pre-clinical efficacy and synergistic potential of the MDM2-p53 antagonists, Nutlin-3 and RG7388, as single agents and in combined treatment with cisplatin in ovarian cancer”. In: *Oncotarget* 7.26 (2016), pp. 40115–40134. DOI: 10.18632/oncotarget.9499.

[498] K. Kumamoto, E. A. Spillare, K. Fujita, I. Horikawa, T. Yamashita, E. Appella, M. Nagashima, S. Takenoshita, J. Yokota, and C. C. Harris. “Nutlin-3a Activates p53 to Both Down-regulate Inhibitor of Growth 2 and Up-regulate mir-34a, mir-34b, and mir-34c Expression, and Induce Senescence”. In: *Cancer Research* 68.9 (2008), pp. 3193–3203. DOI: 10.1158/0008-5472.can-07-2780.

[499] C.-E. Wu, T. S. Koay, A. Esfandiari, Y.-H. Ho, P. Lovat, and J. Lunec. “ATM dependent DUSP6 modulation of p53 involved in synergistic targeting of MAPK and p53 pathways with trametinib and MDM2 inhibitors in cutaneous melanoma”. In: *Cancers* 11.1 (2018), p. 3. DOI: 10.3390/cancers11010003.

[500] K. C. Schreck, A. N. Allen, J. Wang, and C. A. Pratilas. “Combination MEK and mTOR inhibitor therapy is active in models of glioblastoma”. In: *Neuro-Oncology Advances* 2.1 (2020), vdaa138. DOI: 10.1093/noajnl/vdaa138.

[501] M. C. Da Vià, A. G. Solimando, A. Garitano-Trojaola, S. Barrio, U. Munawar, S. Strifler, L. Haertle, N. Rhodes, E. Teufel, C. Vogt, et al. “CIC mutation as a molecular mechanism of acquired resistance to combined BRAF-MEK inhibition in extramedullary multiple myeloma with central nervous system involvement”.

In: *The oncologist* 25.2 (2020), pp. 112–118. DOI: 10.1634/theoncologist.2019-0356.

[502] F. Coussy, R. El-Botty, S. Château-Joubert, A. Dahmani, E. Montaudon, S. Leboucher, L. Morisset, P. Painsec, L. Sourd, L. Huguet, et al. "BRCAness, SLFN11, and RB1 loss predict response to topoisomerase I inhibitors in triple-negative breast cancers". In: *Science Translational Medicine* 12.531 (2020), eaax2625. DOI: 10.1126/scitranslmed.aax2625.

[503] J. Krushkal, Y. Zhao, C. Hose, A. Monks, J. H. Doroshow, and R. Simon. "Longitudinal transcriptional response of glycosylation-related genes, regulators, and targets in cancer cell lines treated with 11 antitumor agents". In: *Cancer Informatics* 16 (2017), p. 1176935117747259. DOI: 10.1177/1176935117747259.

[504] S. Xing, Y. Wang, K. Hu, F. Wang, T. Sun, and Q. Li. "WGCNA reveals key gene modules regulated by the combined treatment of colon cancer with PHY906 and CPT11". In: *Bioscience reports* 40.9 (2020), BSR20200935. DOI: 10.1042/bsr20200935.

[505] I. Iacobucci, N. Iraci, M. Messina, A. Lonetti, S. Chiaretti, E. Valli, A. Ferrari, C. Papayannidis, F. Paoloni, A. Vitale, et al. "IKAROS deletions dictate a unique gene expression signature in patients with adult B-cell acute lymphoblastic leukemia". In: *PloS one* 7.7 (2012), e40934. DOI: 10.1371/journal.pone.0040934.

[506] Z. Tan, J. Zhao, and Y. Jiang. "MiR-634 sensitizes glioma cells to temozolomide by targeting CYR 61 through Raf-ERK signaling pathway". In: *Cancer Medicine* 7.3 (2018), pp. 913–921. DOI: 10.1002/cam4.1351.

[507] X. Ge, M.-H. Pan, L. Wang, W. Li, C. Jiang, J. He, K. Abouzid, L.-Z. Liu, Z. Shi, and B.-H. Jiang. "Hypoxia-mediated mitochondria apoptosis inhibition induces temozolomide treatment resistance through miR-26a/Bad/Bax axis". In: *Cell death & disease* 9.11 (2018), p. 1128. DOI: 10.1038/s41419-018-1176-7.

[508] W. N. Venables and B. D. Ripley. *Modern applied statistics with S*. 4th edition. Springer Science & Business Media, 2002. DOI: 10.1007/978-0-387-21706-2.