Heinrich Heine
Universität
Düsseldorf

Drug response prediction with multi-output machine learning methods: pitfalls and new directions

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Nguyen Khoa Tran
aus Hildesheim

Düsseldorf, Oktober 2025

Gedruckt mit der Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Gunnar W. Klau

2. Prof. Dr. Oliver Ebenhöh

Tag der mündlichen Prüfung: 02.02.2026

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my doctoral supervisor Prof. Dr. Gunnar W. Klau. You created a fun and relaxed environment where I felt genuinely happy. It is incredibly rare to meet someone so accomplished who remains so humble and thoughtful toward their employees. I will always remember you, not only as my doctoral supervisor, but also as a friend.

Next, I would like to thank my second assessor Prof. Dr. Oliver Ebenhöh and all the other assessors of this dissertation: Prof. Dr. Alexander Dilthey, Prof. Dr. Tobias Marschall, and Prof. Dr. Melanie Schmidt.

Furthermore, I owe special thanks to my (former) colleagues Eline van Mantgem, Sven Schrinner, Philipp Spohr, Max Ried, Laura Kühle, Sarah Schweier, Daniel Schmidt, Sara Schulte, Jonathan Bobak, My Ky Huynh, Martin Jürgens, Jan van Grimbergen, and Rebecca Serra Mari. Having you as colleagues is a real blessing. Thanks to you, I was able to enjoy every day at work to its fullest!

Not to be overlooked, although you were not my direct colleagues, I would still like to sincerely convey my thanks you: Angela Rennwanz, Claudia Forstinger, Yannick Schmitz, Martin Breuer, Nan Qin, Martin Papenberg, Janine Golov, Alexander Kroll, Gaia Gigante, Jan Meisner, Jan Meissner, Melanie Schmidt's group, Tobias Marschall's group, Alexander Dilthey's group as well as all the university (hospital) staff and students I worked with. All of you have made my time working at the university even more enjoyable.

Finally, I extend my heartfelt thanks to my husband, family, friends as well as everyone and everything, past and present, that has influenced my life. Thank you for allowing me to keep growing!

Abstract

Predicting the effect of anti-cancer drugs on cancer cells is essential to understanding biological and chemical mechanisms relevant to cancer treatment. Current drug response prediction methods are typically machine learning models trained on two inputs, multi-omics features of cancer cell lines and chemical features of drugs, to predict outputs, commonly represented by drug response metrics such as IC_{50} or AUC. In this dissertation, limitations of existing methods as well as currently used data and metrics are discussed, and shifts in research focus are suggested.

First, it is explored whether multi-output support vector regression can capture correlations across different outputs. The findings indicate that support vector regression is unsuitable for this task, in contrast to artificial neural networks.

Next, the performance of neural networks on the large-scale cancer research dataset from the DepMap project is investigated. Particularly, TGSA is examined, the currently leading deep learning method. While TGSA sometimes learns correlations between outputs, these correlations lack biological and chemical relevance. Furthermore, TGSA does not always exceed baseline performance, and when it does, it still fails to surpass a simple multilayer perceptron. These deficiencies are attributed to data inconsistencies in both input and output data as well as unsuited modeling of pharmacodynamics, and alternative modeling approaches are proposed.

As an initial step toward addressing the deficiencies regarding output data, an alternative to traditional drug response metrics like IC_{50} and AUC, derived from dose-response curves, is needed. Dose-response curves are modeled as 4PL curves, which rely on relatively few measurements and are prone to instability. Live-cell imaging data are images of cell cultures captured at user-defined time intervals (e.g., every 15 minutes), adding a time dimension to increase the number of measurements, improving model stability. By combining the 4PL curve with a logistic function, a new model, VUScope, is introduced to fit dose-time-response surfaces. Along with VUScope, a new drug response metric, GRIVUS, is proposed to replace IC_{50} and AUC in the long run as GRIVUS allows for more equitable comparisons across different cell lines and drugs. VUScope also enables the prediction of long-term drug responses based on short-term data and still yields reliable results when applied to datasets with few time points (e.g., measurements taken every 24 hours). This makes VUScope compatible with traditional HTS data, enabling labs without live-cell imaging systems to use VUScope by taking measurements every 24 hours. Moderate to high correlations can be observed between drug response results obtained from live-cell imaging and traditional HTS data.

In conclusion, pitfalls in current drug response prediction research are identified and new directions are outlined, providing valuable insights for both wet-lab and dry-lab experiments aimed at advancing cancer treatment.

List of publications

Primary publications

- NK Tran, LC Kühle, GW Klau
  A critical review of multi-output support vector regression
  Published in Elsevier Pattern Recognition Letters, 2023
  https://doi.org/10.1016/j.patrec.2023.12.007

- NK Tran, GW Klau
  Drug response prediction: A critical systematic review of current datasets and methods
  Published in Elsevier Pattern Recognition Letters, 2025
  https://doi.org/10.1016/j.patrec.2025.10.016

- NK Tran*, MK Huynh*, AD Kotman, M Jürgens, T Kurz, S Dietrich, GW Klau†, N Qin†
  VUScope: a mathematical model for evaluating image-based drug response measurements and predicting long-term incubation outcomes
  Published in Bioinformatics, 2026
  https://doi.org/10.1093/bioinformatics/btaf679

Other publications

- M Brand*, NK Tran*, P Spohr, S Schrinner, GW Klau
  The Homo-Edit Distance Problem
  Uploaded on bioRxiv, 2020
  https://doi.org/10.1101/2020.05.27.118273

- M Papenberg, M Breuer, M Diekhoff, NK Tran, GW Klau
  Extending the Bicriterion Approach for Anticlustering: Exact and Hybrid Approaches
  Published in Psychometrika, 2025
  https://doi.org/10.1017/psy.2025.10052

- NK Tran, L Mu, M Papenberg, GW Klau
  The 2-coloring problem with size constraints
  To be submitted to European Symposium on Algorithms (ESA), 2026

- J Meissner, F Nellen, NK Tran, G Rauhut, GW Klau, J Meisner
  Ro-Vibrational Spectroscopy of Systematically Generated Interstellar Molecules
  To be submitted to Journal of Chemical Information and Modeling, 2026

*Shared first authors
†Shared last authors

Contents

List of abbreviations

3D	=	Three-dimensional
4PL	=	4-parameter logistic
AUC	=	Area under the dose-response curve
CCLE	=	Cancer Cell Line Encyclopedia
CN	=	Copy number
CNV	=	Copy number variation
DepMap	=	Dependency Map
EC_{50}	=	Half-maximal effective concentration
EXP	=	Gene expression
FPKM	=	Fragments per kilobase of transcript per million reads mapped
GDSC	=	Genomics of Drug Sensitivity in Cancer
GNN	=	Graph neural network
GR	=	Growth rate inhibition
GRIVUS	=	Growth rate inhibition volume under the dose-time-response surface
HTS	=	High-throughput drug screening
IC_{50}	=	Half-maximal inhibitory concentration
MLP	=	Multilayer perceptron
MMLP	=	Multi-output multilayer perceptron
MUT	=	Somatic mutation
NN	=	Neural network
PPI	=	Protein-protein interaction
RF	=	Random forest
SMILES	=	Simplified molecular input line entry system
SNV	=	Single nucleotide variant
SV	=	Structural variant
SVR	=	Support vector regression
TGSA	=	Twin Graph Neural Networks with Similarity Augmentation
TPM	=	Transcript per million
WES	=	Whole-exome sequencing
WGS	=	Whole-genome sequencing

Chapter 1

Introduction

Drug response prediction is a computational task that aims to estimate how cancer cells respond to various therapeutic drugs. Specifically, it involves predicting quantitative measures of drug efficacy based on cell line and drug features.

The importance of drug response prediction stems from its potential to support personalized medicine. By identifying which treatments are likely to be effective for specific patients based on their gene profiles, costs and risks associated with trial-and-error approaches can be reduced, ultimately improving patient outcomes. In addition, drug response prediction models may capture information about underlying biological and chemical mechanisms, further contributing to the advancement of precision oncology.

To build such models, researchers use the Dependency Map (DepMap) project [1], which includes many large-scale cancer research projects such as the Cancer Cell Line Encyclopedia (CCLE) project [2], from which DepMap originally emerged, as well as the Genomics of Drug Sensitivity in Cancer (GDSC) project [3] and Cell Model Passports [4]. DepMap provides comprehensive molecular characterizations of cancer cell lines and reports drug response measurements, which serve as target variables for model training. With the included drug names or PubChem identifiers [5], chemical features (e.g., Lewis structures) can be obtained.

Drugs with similar mechanisms of action or structural features may elicit similar effects on cell lines. Accounting for these correlations could significantly improve prediction accuracy and generalizability. Therefore, predicting responses to multiple drugs simultaneously using multi-output models is preferable to treating drugs independently.

As many machine learning models are inherently capable of multi-output prediction and recent advances in artificial intelligence have revolutionized numerous fields, techniques such as ensemble models, kernel methods, and deep learning architectures have emerged as state-of-the-art methods for drug response prediction. These methods, while still being explored, show promise in computational oncology due to their ability to model non-linear relationships, handle various data types, and uncover hidden patterns.

This dissertation contributes to the field of drug response prediction by briefly summarizing key concepts, critically evaluating current datasets and machine learning models, and introducing the first dose-time-dependent drug response metric derived from a new dose-time-response model. In the end, the main findings of the primary publications of this cumulative dissertation are revisited and additional discussion points are addressed, followed by closing remarks to guide future drug response prediction research.

Chapter 2

Background

This chapter outlines concepts commonly encountered in drug response prediction.

2.1 Pharmacodynamics

Pharmacodynamics is the branch of pharmacology concerned with biochemical and physiological effects of chemical compounds, also referred to as drugs, on living organisms, including humans, animals, microorganisms and combinations of organisms. On the other hand, pharmacokinetics is the branch that studies how an organism absorbs, distributes, metabolizes, and eventually excretes a drug.

Most drugs are ligands, i.e., drugs that exert their effects by binding to specific targets such as receptors (proteins, enzymes) or RNA molecules, typically inducing structural changes in the target, leading to altered function [6]. This functional alteration generally falls into two categories: the activation of a biological response or the inhibition or suppression of that response. In contrast, chemically reactive drugs do not act by binding to specific targets. Instead, they react directly with cellular components like membranes or DNA. Besides ligands and chemically reactive drugs, there are other types of drugs, including prodrugs, which require metabolic conversion by the organism (a process studied in pharmacokinetics), as well as drugs whose mechanisms of action remain unclear. Biological responses to drugs can encompass a wide range of outcomes, including triggering apoptosis or other forms of cell death, inhibition or stimulation of proliferation, modulation of signaling pathways, altered gene or protein expression, and in some cases no measurable effect.

A key focus of pharmacodynamics is the dose-response relationship that can be described by the two concepts of potency and efficacy, see Figure 2.1. Potency reflects the amount of drug needed to produce a specific effect (e.g., 50% of the maximum response) on an organism, with high potency meaning a low dose is sufficient. High potency may reduce the risk of sensitization, simplify administration, and significantly lower production costs, but it is of secondary importance to patient outcomes and may increase the risk of overdose [7]. Efficacy denotes the maximum response a drug can produce on an organism regardless of dose [8], with high efficacy being desirable. Note that although dose refers to the amount of administered drug (e.g., 200 mg) in in vivo (whole-organism) studies and concentration refers to the amount of drug per unit volume (e.g., 1 μM) in in vitro (laboratory) studies, the term dose-response is commonly used in both contexts.

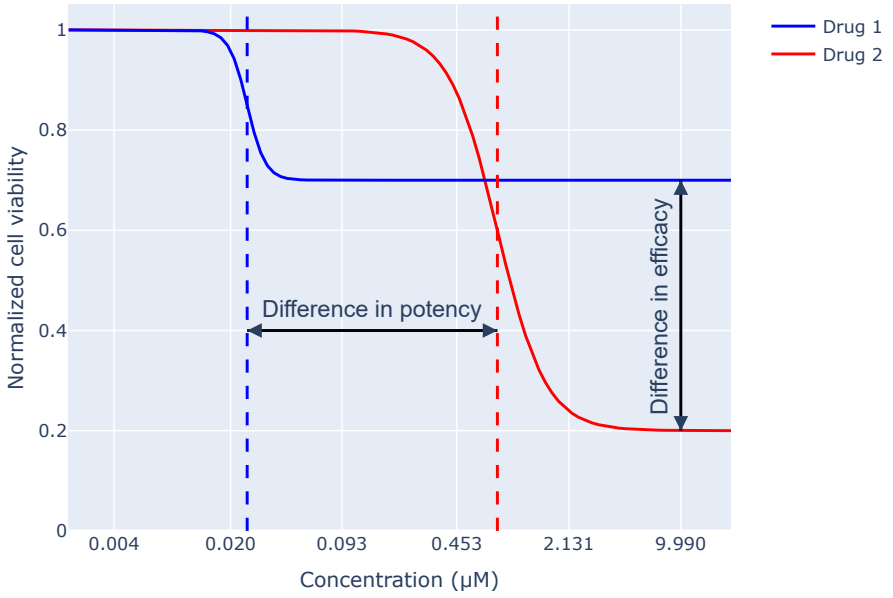


Figure 2.1: Potency and efficacy. Two drugs are administered to the same cell line. For this cell line, Drug 1 is more potent, producing a response at a lower concentration, whereas Drug 2 is more efficacious, producing a greater maximal response as the concentration approaches ∞. The blue and red dashed lines indicate the concentrations at which 50% of the maximal response is achieved for Drug 1 and Drug 2, respectively.

2.2 Drug screening and assays

Drug screening is the process of experimentally applying drugs to biological systems to facilitate drug discovery, i.e., to identify drugs that produce desirable effects such as killing cancer cells or inhibiting disease-related activity. The goal is to generate biological response data that can later be analyzed to determine how drugs affect living organisms. Figure 2.2 shows both drug screening system presented in Sections 2.2.1 and 2.2.2.

Assays are divided into two types: endpoint assays, which involve taking a single measurement by lysing and killing the cells, and kinetic assays, which allow for repeated, non-lethal measurements over time.

2.2.1 High-throughput drug screening

High-throughput drug screening (HTS) is a rapid, automated method that can be paired with either endpoint or kinetic assays to test large numbers of drugs for their effects on specific targets or pathways. Drugs are typically tested at multiple concentrations to evaluate dose-dependent effects, aiding in identifying appropriate doses for patients.

HTS in research laboratories is mainly performed using 384- or 1536-well plate formats. In these experiments, a variety of drugs are randomly allocated across the plate, usually at 5 to 8 different concentrations. To ensure the quality of assay results, each drug plate also contains vehicle controls. These controls help calculate solvent toxicity, serve as normalization standards, and allow for assessment of assay variability and reproducibility. Positive controls are included as proof of principle, as well as to determine the assay's dynamic range and window. See Figure 2.2A for an image of an HTS system.

2.2.2 Live-cell imaging

Live-cell imaging is a drug screening method that involves continuously capturing images to monitor cellular behavior over time, making live-cell imaging a kinetic assay. While kinetic assays for HTS capture only aggregated measurements per well, live-cell imaging tracks individual cells over time, providing detailed single-cell information. Like HTS experiments, live-cell imaging experiments are conducted with plates, although their throughput is considerably lower.

The Incucyte SX5 by Sartorius stands out among live-cell imaging systems due to its ability to image six 384-well plates simultaneously. Housed within an incubator, it maintains a stable environment with controlled temperature, CO_2 levels, and humidity, all of which are essential for long-term experiments. This makes it well-suited for semi-high-throughput and extended live-cell imaging studies. Furthermore, it is equipped with three lasers, enabling the application of various fluorescent dyes to analyze cell characteristics, for instance, employing cytotoxic dyes to differentiate between live and dead cells, and using Annexin V and Caspase 3 dyes to identify early and late apoptosis. See Figure 2.2B for an image of a live-cell imaging system.

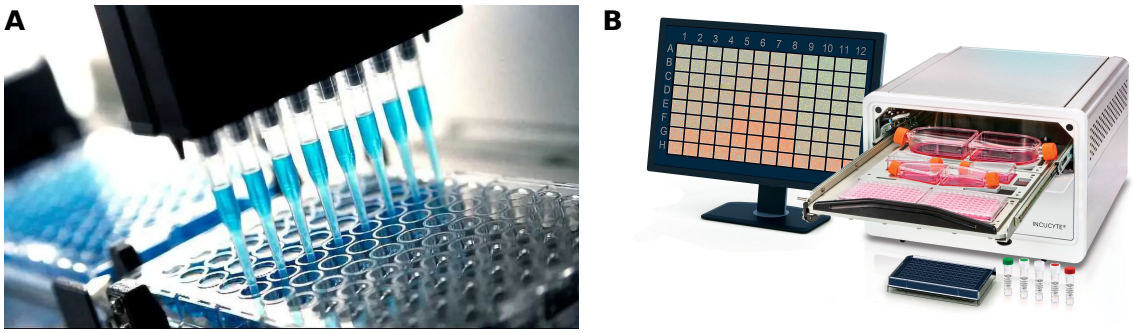


Figure 2.2: High-throughput drug screening and live-cell imaging. Panel A shows test samples being automatically pipetted into a grid of wells in a plate. Image taken from [9]. Panel B shows the Incucyte system by Sartorius next to a monitor displaying images of a grid of wells in a plate at a single time point. Image taken from [10].

2.2.3 Cell viability assays

Cell viability assays are endpoint assays that can be used in HTS experiments to determine the proportion of live cells by detecting biochemical differences between live and dead cells. For instance, the CellTiter-Glo (CTG) assay [11] quantifies viable cells based on intracellular adenosine triphosphate (ATP) levels. To release ATP, a detergent in the CTG reagent lyses and thereby kills the cells. The released ATP then serves as a substrate for a luciferase reaction, in which luciferin is converted to oxyluciferin, producing light proportional to the number of metabolically active cells. Subsequently, the luminescence can be measured with a luminometer such as GloMax [12] and then quantified. For example, in GDSC [13] raw luminescence values are scaled relative to a positive control (i.e., wells without cells) as well as a negative control (i.e., wells with untreated cells), representing 0% or 100% viability, respectively. Values above 1, occurring due to experimental inaccuracies or drugs stimulating cell proliferation, are capped at 1, resulting in normalized cell viability values ranging from 0 to 1. Figure 2.3 illustrates the CTG assay.

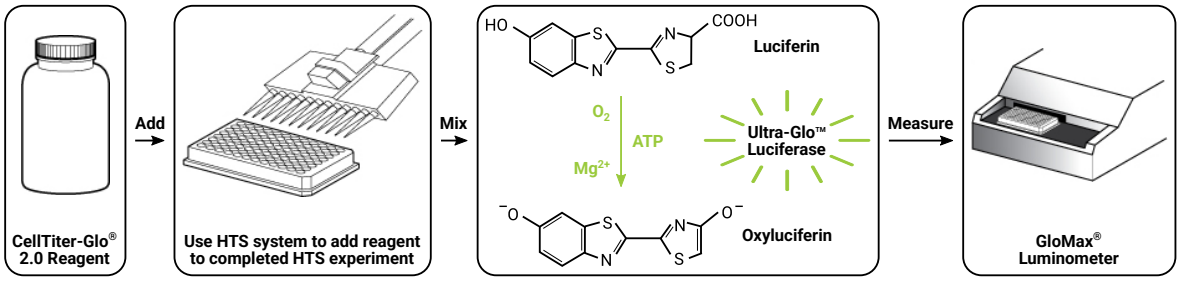


Figure 2.3: CellTiter-Glo assay. After completion of an HTS experiment, an HTS system may add the reagent to the plate and mix it thoroughly to ensure uniform cell lysis. ATP released from the lysed cells causes a luciferase reaction that produces light, which can then be measured with a luminometer. Finally, these measurements can be quantified relative to positive and negative controls, yielding cell viability values between 0 and 1. Images adapted from [14].

2.3 4-parameter logistic curve/E_{max} model and Hill equation

Given cell viability measurements, the 4-parameter logistic (4PL) curve [15], also known as the E_{max} model [16], is a commonly used sigmoidal function fitted to the measurements to model dose-response relationships. The cell viability f at a concentration $d' \in \mathbb{R}$, $d' \geq 0$, is modeled as

$$f(d') = \frac{\alpha - \delta}{1 + \left(\frac{d'}{\gamma'}\right)^{\beta}} + \delta.$$

Drug concentrations are typically spaced exponentially rather than linearly to efficiently cover a broader range, and are therefore expressed on a logarithmic scale. For a $\log_{10}$-transformed concentration $d = \log_{10}(d')$, the 4PL curve becomes

$$f(d) = \frac{\alpha - \delta}{1 + 10^{\beta \cdot (d - \gamma)}} + \delta.$$

Figure 2.4 shows an example dose-response curve. In this model, $\alpha \in \mathbb{R}$ and $\delta \in \mathbb{R}$ represent the asymptotic cell viability values to which the curve converges as $d \to -\infty$ and $d \to \infty$, respectively. The inflection point, corresponding to the concentration at which 50% of the maximum response is achieved, is described by $\gamma = \log_{10}(\gamma')$, $\gamma \in \mathbb{R}$. The steepness parameter $\beta \in \mathbb{R}$, $\beta \geq 0$ reflects the sensitivity of cell viability to changes in concentration near the inflection point. The four parameters are estimated by a fitting algorithm such as the trust region reflective algorithm [17] used by the `least_squares` function in SciPy [18], which performs optimization with respect to a chosen metric, for example, the mean absolute percentage error. The difference between the 4PL curve and the Hill equation [19] is that the latter has fixed asymptotes, with $\alpha = 1$ and $\delta = 0$.

Note that in the original 4PL curve, β is not restricted to be strictly positive. However, if $\beta < 0$, the roles of α and δ are reversed such that the curve approaches α as $d \to \infty$ and δ as $d \to -\infty$. Therefore, to maintain the roles of α and δ and additionally reduce the solution space for β, the restriction $\beta \geq 0$ is imposed.

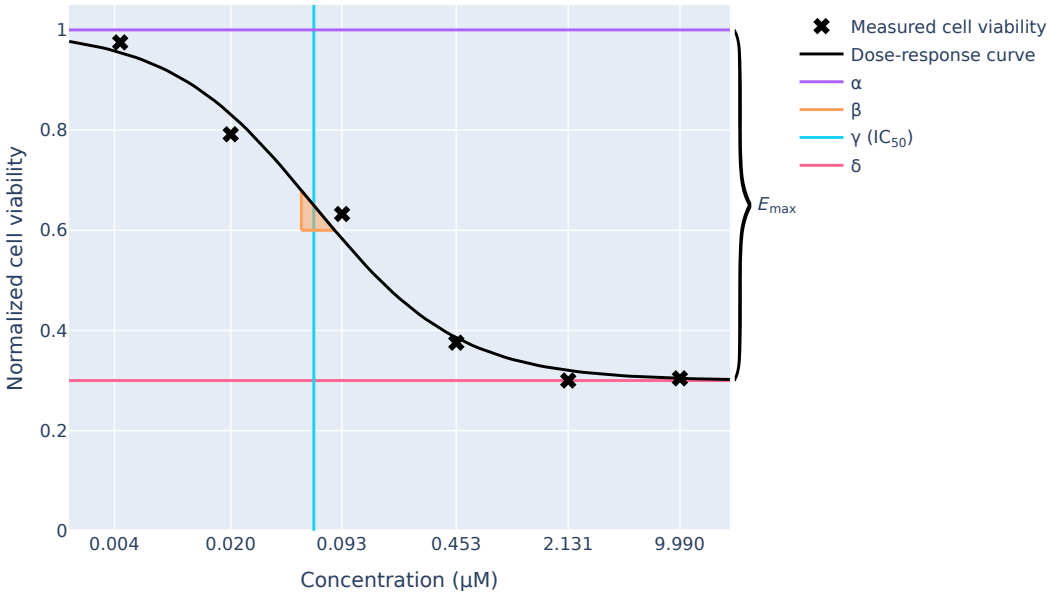


Figure 2.4: Dose-response curve. A 4PL curve is fitted to six measured normalized cell viability values at six different concentrations spaced on a logarithmic scale, with indicated asymptotes α and δ, steepness parameter β, inflection point γ (corresponding to the IC_{50}, see Section 2.4.1), and E_{max} (see Section 2.4.2).

2.4 Experimental output data

The following metrics derived from dose-response curves are commonly used as prediction targets for current drug response prediction models.

2.4.1 Half-maximal inhibitory concentration and half-maximal effective concentration

The half-maximal inhibitory concentration (IC_{50}) is the concentration needed to inhibit 50% of a specific biological or biochemical function. Conversely, the half-maximal effective concentration (EC_{50}) is the concentration needed to activate 50% of such a function. Both IC_{50} and EC_{50} are metrics for potency, not for efficacy.

The inflection point γ from the 4PL curve corresponds to either the IC_{50} (monotonically decreasing curve) or EC_{50} (monotonically increasing curve). Both GDSC and CCLE report only IC_{50} values, as there appear to be no growth-stimulating cases in their datasets. Note that if the curve fit algorithm estimates the inflection point γ to be greater than the maximum tested concentration, CCLE reports IC_{50} values capped at the maximum tested concentration, while GDSC keeps the estimated IC_{50} values.

2.4.2 E_{max}

The E_{max} represents the efficacy of the drug and corresponds to α minus δ from the 4PL curve. Unlike IC_{50} or EC_{50}, it is not a metric for potency. CCLE reports E_{max} under the name A_{max} [20], while GDSC does not report E_{max}.

2.4.3 Area under the dose-response curve

The area under the dose-response curve (AUC) quantifies the overall drug response across the tested concentration range. As AUC accounts for the entire dose-response curve, it is less dependent on precise viability measurements [21] near the inflection point than IC_{50} or near the concentration limits than E_{max}. AUC can also be approximated directly from measured cell viability values, eliminating the need for curve fitting, which is especially useful when curve fits are unreliable due to assays with fewer than five concentrations [22]. The main advantage of AUC is that it partially captures both potency and efficacy in a single metric [23]. Low AUC values indicate that a drug is both potent and efficacious. Medium values suggest it is either potent but less efficacious, efficacious but less potent, or moderate in both. High values indicate low efficacy and possibly low potency. Although the interpretation of medium and high AUC values is imprecise, only low values are typically of primary interest. A key limitation, however, is its dose dependence: The same dose-response curve has varying AUC values for different concentration ranges, making comparisons across experiments questionable; even within the same range, comparing drugs with substantially different dosing recommendations, i.e., different inflection points, remains problematic.

The AUC is computed as the integral of the dose-response curve from the minimum to the maximum tested concentration, optionally followed by a normalization. Figure 2.5 illustrates the normalization process. The advantage of the normalization is interpretability: Normalized AUC values can be interpreted as percentages, where values indicate stronger inhibitory effects the closer they are to 0, values around 1 correspond to no effect, and values above 1 suggest growth-stimulating effects. Note that the term AUC commonly refers to the normalized AUC as in GDSC [13]. CCLE reports the activity area (ActArea) instead of the AUC, which corresponds to a non-normalized area over the dose-response curve with $y = 1$ as the upper bound [20].

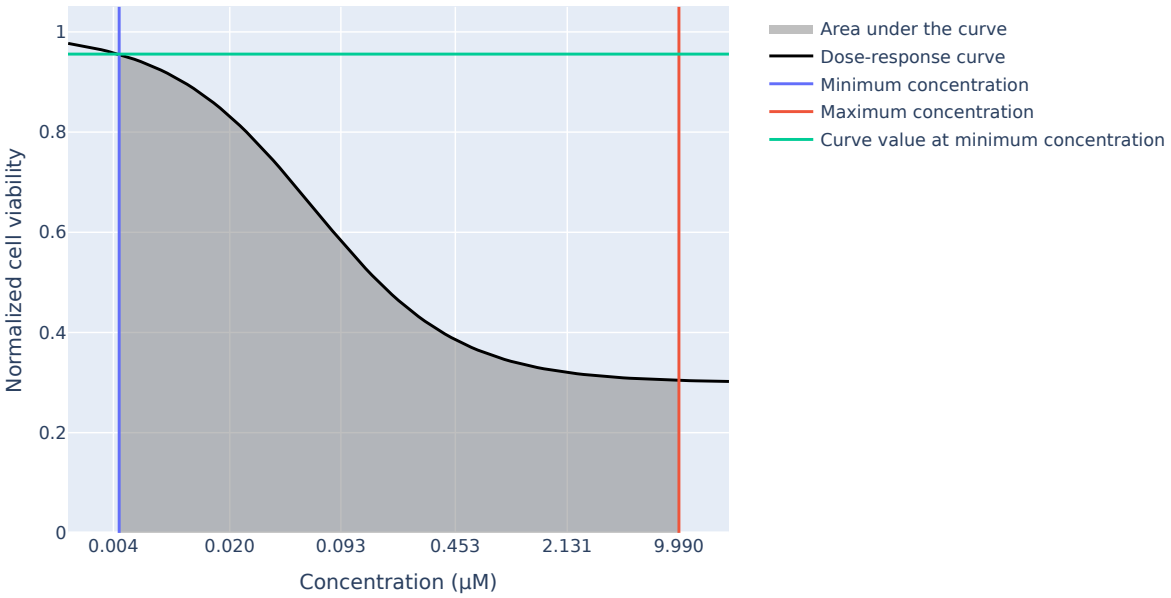


Figure 2.5: Visualization of raw and normalized AUC. The raw AUC corresponds to the gray area. To obtain the (normalized) AUC, the raw AUC is divided by the area of the rectangle defined by the red line, blue line, green line, and the (logarithmic) x-axis.

2.4.4 Growth rate inhibition

IC_{50}, E_{max}, and AUC are influenced by the growth rate, or more precisely, the cell proliferation rate, making them incomparable across different cell lines [22]. To address this limitation, Hafner et al. [22] introduced the growth rate inhibition (GR) approach. For endpoint assays, GR consists of three components, outlined below and additionally illustrated in Figure 2.6.

Before describing these components, it is useful to define the underlying growth model. Although not explicitly stated in the original paper, the surrounding equations suggest the growth model can be written as follows: The cell count or viability, referred to as the cell number in the following, at concentration d and time t is given by the growth model $x(d,t) = x_0 \cdot e^{k(d)\cdot t}$, where x_0 is the initial cell number and $k(d)$ is the doubling rate when treated at concentration d. The untreated cell division time is denoted as $t_2 = \frac{\ln(2)}{k(0)}$.

1. The first component is the GR value. It is defined as

$$\mathrm{GR}(d) = 2^{\frac{k(d)}{k(0)}} - 1,$$

where the ratio $\frac{k(d)}{k(0)}$ reflects the relative (rather than absolute) change in doubling rate, thereby removing the influence of differing proliferation rates. Note that although the original paper describes $\mathrm{GR}(d)$ as a ratio normalized to one cell division, only the exponent $\frac{k(d)}{k(0)}$ represents that ratio. The complete expression is a nonlinear transformation that maps this ratio to the range $[-1, \infty)$, with the following interpretation:

- $\mathrm{GR}(d) < 0$: cytotoxic effect, i.e., net cell loss,
- $\mathrm{GR}(d) = 0$: cytostatic effect, i.e., complete growth inhibition,
- $0 < \mathrm{GR}(d) < 1$: partial growth inhibition,
- $\mathrm{GR}(d) = 1$: no drug effect,
- $\mathrm{GR}(d) > 1$: growth stimulation (omitted in the original paper, which disregards growth-stimulating drugs).

As $k(d)$ is typically unknown, the GR value can instead be computed based on the incubation time t and measured cell numbers at the end of incubation $x(d,t)$ and $x(0,t)$. This can be done in one of two ways: If the initial cell number x_0 is known, it can be calculated as

$$\mathrm{GR}(d) = 2^{\frac{\log_2\left(\frac{x(d,t)}{x_0}\right)}{\log_2\left(\frac{x(0,t)}{x_0}\right)}} - 1,$$

or alternatively, if the untreated division time t_2 is known, as

$$\mathrm{GR}(d) = 2^{1+\frac{\log_2\left(\frac{x(d,t)}{x(0,t)}\right)}{\frac{t}{t_2}}} - 1.$$

Note that neither GDSC nor CCLE provide initial cell counts nor cell division times for GR to be computed.

2. The second component is the GR curve fitting model. It is defined as

$$\mathrm{GR}_{\mathrm{curve}}(d) = \mathrm{GR}(\infty) + \frac{1 - \mathrm{GR}(\infty)}{1 + \left(\frac{d}{\gamma'}\right)^{\beta}}.$$

   It differs from the 4PL curve or Hill equation in that its range is $[-1, 1]$, rather than $[0, 1]$. Note that in the original paper, both $\mathrm{GR}(d)$ and $\mathrm{GR}_{\mathrm{curve}}(d)$ are named $\mathrm{GR}(d)$.

3. The third component consists of the GR metrics. They can be derived from the fitted GR curve and are defined as follows:

   - GR_{50} is either set to the concentration d at which $\mathrm{GR}_{\mathrm{curve}}(d) = 0.5$ or set to ∞ if $\forall d : \mathrm{GR}_{\mathrm{curve}}(d) > 0.5$. Its underlying concept is inspired by the IC_{50}, but unlike the IC_{50}, it does not necessarily correspond to the inflection point, which would be γ'. In fact, in most cases, $\mathrm{GR}_{\mathrm{curve}}(\gamma') \neq 0.5$, and the IC_{50} is also not set to ∞ if $\forall d : \mathrm{GR}_{\mathrm{curve}}(d) > 0.5$.
   - $\mathrm{GR}_{\mathrm{max}}$ is the value of $\mathrm{GR}_{\mathrm{curve}}(d)$ at the highest tested drug concentration. It is conceptually related to E_{max}, but differs in definition because E_{max} corresponds to $1 - \mathrm{GR}(\infty)$.
   - $\mathrm{GR}_{\mathrm{AOC}}$ is the area over the GR curve (AOC) across the tested concentration range. It is conceptually related to the AUC, the latter being computed using $y = 0$ as the lower bound. In contrast, since the GR curve can take negative values, the AOC is instead computed using $y = 1$ as the upper bound.

These definitions are adapted from the original paper and expressed in notation consistent with the 4PL curve described in Section 2.3. The original paper briefly mentions two extensions of the GR value to incorporate time dependence for use in kinetic assays:

$$\mathrm{GR}(d, t) = 2^{\frac{\log_2\left(\frac{x(d,t)}{x_0}\right)}{\log_2\left(\frac{x(0,t)}{x_0}\right)}} - 1, \text{ and}$$

$$\mathrm{GR}(d, t, \Delta t) = 2^{\frac{\log_2\left(\frac{x(d,t+\Delta t)}{x(d,t-\Delta t)}\right)}{\log_2\left(\frac{x(0,t+\Delta t)}{x(0,t-\Delta t)}\right)}} - 1.$$

However, both expressions are mathematically equivalent to the time-independent GR value $\mathrm{GR}(d)$. Substituting the corresponding expressions for x shows that the time terms t and Δt cancel out in both the numerator and denominator. This also holds true when the growth model from the original paper,

$$x(d, t) = x_0 \cdot e^{t \cdot k(0) \cdot \left(1 - \frac{S_M \cdot d^{\beta}}{\mathrm{SC}_{50}^{\beta} + d^{\beta}}\right) - t \cdot \frac{k_L \cdot d^{\beta}}{\mathrm{LC}_{50}^{\beta} + d^{\beta}}}$$

(where the meanings of S_M, SC_{50}, LC_{50}, and k_L are not relevant in this context), is used in place of the assumed equivalent growth model, $x(d, t) = x_0 \cdot e^{k(d) \cdot t}$, see Section 3.3.2, Appendix A. Note that in the original paper by Hafner et al., both $\mathrm{GR}(d, t)$ and $\mathrm{GR}(d, t, \Delta t)$ are named $\mathrm{GR}(d, t)$.

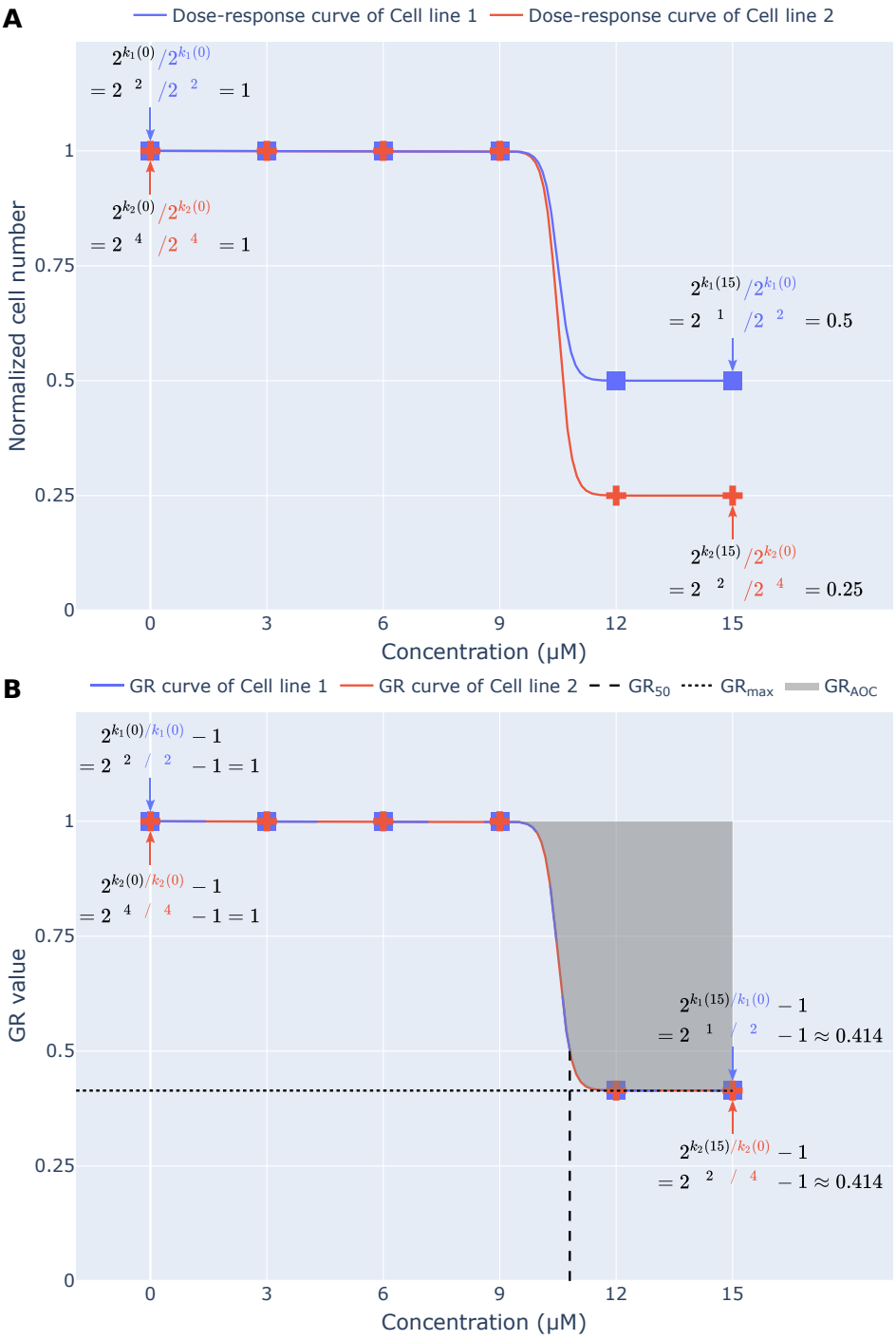


Figure 2.6: Traditional dose-response model vs. growth rate inhibition model. Two cell lines with initial growth rates $k_1(0) = 2$ and $k_2(0) = 4$ were tested with drug concentrations from 0 to 15 μM (step size 3 μM). The drug effect is identical on both cell lines: Up to 9 μM, both growth rates were unaffected; from 12 μM onward, both were halved. In Panel A, the two dose-response curves (4PL) computed from normalized cell numbers, obtained by dividing all cell counts by the initial cell count $2^{k(0)}$, are different, despite the identical drug effect on both cell lines. In Panel B, the two GR curves computed from GR values, obtained by dividing all growth rates by the initial growth rate $k(0)$, are identical, consistent with the identical drug effect on both cell lines. Furthermore, all three GR metrics are indicated: GR_{50} (dashed line), GR_{max} (dotted line), and GR_{AOC} (gray area).

2.5 Biological input data

The following cell line-derived data types are commonly considered as input features for current drug response prediction models. Note that while both GDSC and Cell Model Passports are associated with DepMap, they have their own dedicated download pages, which are not directly accessible through the DepMap portal [24], whereas the dataset DepMap Public refers specifically to the CRISPR and omics data accessible through the DepMap portal. Figure 2.7 summarizes structural variants (SVs) from Sections 2.5.1 and 2.5.2, while Figure 2.9 summarizes the process from gene to protein, connecting the contents of Sections 2.5.1 through 2.5.5.

2.5.1 Somatic mutations

A somatic mutation (MUT) is a type of SV that refers to a genomic alteration in the DNA sequence that occurs in non-germline cells, meaning it is generally not inherited. Exceptions exist in plants, which do not have a germline, and asexually reproducing organisms. MUT data include single nucleotide variants (SNVs), small insertions and deletions (indels), and occasionally more complex SVs such as translocations and inversions, see Figure 2.7A.

Raw MUT data are commonly identified through whole-exome sequencing (WES) or whole-genome sequencing (WGS), where reads from a sample with SVs are aligned to a reference sequence. Cell Model Passports processes the raw data with CaVEMan [25] or Pindel [26], and DepMap Public processes the raw data with Mutect2 [27] from GATK [28], resulting in MUTs represented as binary features (mutated vs. wild-type) or as categorical features indicating the variant type (e.g., SNV, insertion, deletion). Additional variant information may also be given, e.g., synonymous (encodes the same amino acid), missense (encodes another amino acid), or nonsense (encodes no amino acid). Neither Cell Model Passports nor DepMap Public includes more complex SVs in their MUT data. The GDSC pan-cancer dataset only contains binary features, while Cell Model Passports and DepMap Public datasets contain categorical mutation annotations.

2.5.2 Copy number variations

A copy number variation (CNV) is a type of SV that refers to a genomic alteration involving the duplication or absence of large DNA segments in one or both chromosome copies, which can affect parts of genes, entire genes, or even larger regions, see Figure 2.7B.

For analysis, CNV data are typically represented as per-gene values based on the copy number (CN) state of the DNA region where the gene is located. CN can be reported as either relative or absolute. Relative CN is typically expressed as a $\log_2$ ratio, i.e., the $\log_2$ of the CN at a locus divided by the average CN across the genome for the tissue sample, without the need to estimate tumor purity (proportion of tumor cells in tissue sample) and ploidy. Absolute CN, called total CN in Cell Model Passports but referred to as absolute CN in DepMap Public, is a computed integer CN estimate based on estimated tumor purity and ploidy, which is commonly classified into the following categories: deletion (DNA segment absent in more than one chromosome copy), loss (DNA segment absent in one chromosome copy), neutral (no CNV), gain (few duplications of DNA segment),

amplification (many duplications of DNA segment). The exact threshold at which a gain is classified as an amplification differs between processing methods.

Raw CNV data are commonly inferred from genomic data using single nucleotide polymorphism (SNP) arrays, WES, or WGS. In earlier Cell Model Passports versions, SNP array data are processed with either PICNIC [29] for absolute CNs or GISTIC [30] for categorical GISTIC scores, which can be interpreted according to the previously mentioned categories. In newer Cell Model Passports versions, to obtain absolute CNs, WES data are processed with GATK and PureCN [31], while WGS data are processed with PURPLE [32]. In DepMap Public, both WES and WGS data are processed with either GATK for relative CNs or PureCN (or ABSOLUTE [33] in earlier DepMap Public versions) for absolute CNs. Note that GATK outputs a `MEAN_LOG2_COPY_RATIO`, which corresponds to the above defined ratio of the relative CN including a panel of normals correction [34] through singular value decomposition, median adjustment, and a shift to 1 before the $\log_2$-transformation, and DepMap Public reports $\texttt{Segment\_Mean} = 2^{\texttt{MEAN\_LOG2\_COPY\_RATIO}}$ as their relative CN.

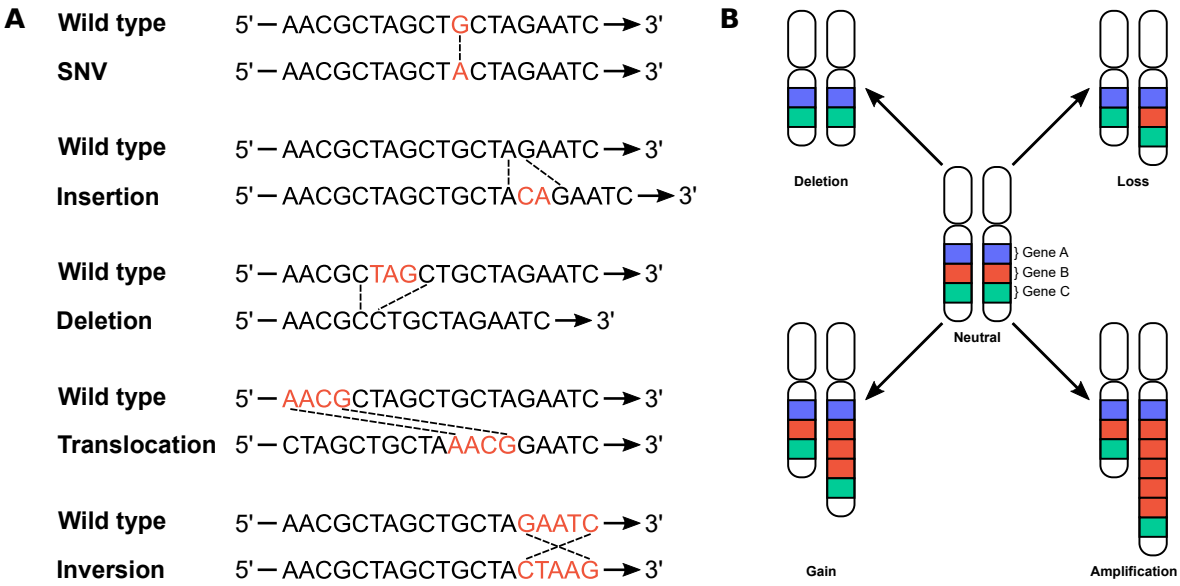


Figure 2.7: Structural variants. Panel A illustrates five types of MUTs in a 5' → 3' DNA strand. The dashed lines serve as visual guides. Panel B illustrates the five categories of CNVs in a chromosome pair. In all examples, only the entire Gene B is affected.

2.5.3 DNA methylation

DNA methylation involves the addition of methyl groups (one carbon atom bonded to three hydrogen atoms, CH_3) to DNA. With few exceptions, this modification primarily occurs on cytosines within 5'-C-phosphate-G-3' (CpG) sites, which is a cytosine followed by a guanine in the 5' → 3' DNA strand. Specifically, methylation typically affects the cytosine on the 5' → 3' strand and the complementary cytosine opposite the following guanine on the 3' → 5' strand. This epigenetic modification influences gene transcription without altering the underlying DNA sequence.

DNA methylation data are typically obtained using array-based platforms such as the Infinium® HumanMethylation450 BeadChip by Illumina [35]. For a DNA region, Cell Model Passports provides raw methylation data as beta values between 0 and 1, indicating the proportion of cells where a methylation occurs within the region.

Current drug response prediction methods do not incorporate DNA methylation data, likely because it is reported for genomic regions in a chromosome rather than for specific genes (e.g., `chr10:100028204-100028508` in GDSC). Summarizing multiple beta values of genomic regions located within a single gene can lead to information loss since methylation can repress or enhance transcription. Moreover, CpG sites can fall between two genes, making it difficult to assign the methylation to a specific gene.

2.5.4 Gene expression

Gene expression (EXP) refers to the production of RNA transcripts from genes. It reflects how actively a gene is being transcribed into mRNA, which serves as an intermediate step in the production of proteins. While many genes encode proteins, others give rise to functional non-coding RNAs with regulatory, structural, or catalytic roles. Because transcript levels do not account for post-transcriptional modifications, translation efficiency, or protein activity, they serve only as proxies for gene function.

EXP values are usually given as raw read counts, fragments per kilobase of transcript per million reads mapped (FPKM) (also called reads per kilobase of transcript per million reads mapped (RPKM)) or transcript per million (TPM). A raw read count q_i is the number of reads that align to gene i, but raw read counts are unsuitable for comparing EXP levels between different tissue samples due to differences in transcript length, library size (total number of reads per sample), and technical artifacts. FPKM corrects for both transcript length l_i and library size $\sum_j q_j$ to enable comparing EXP levels within a single sample. TPM additionally normalizes raw read counts by transcript length such that the sum of all TPMs in each sample equals 10^6 to enable comparing EXP levels across samples. For gene i, FPKM and TPM are defined as:

$$\text{FPKM}_i = \frac{\frac{q_i}{l_i}}{\sum_j q_j} \cdot 10^9 \quad \text{and} \quad \text{TPM}_i = \frac{\frac{q_i}{l_i}}{\sum_j \frac{q_j}{l_j}} \cdot 10^6.$$

However, it has been shown that normalization methods such as trimmed mean of M values (TMM) [36] and the median-of-ratios from the DESeq2 R package [37], applied to raw read counts, produce EXP values that are robust to variations in library size and composition, making them suitable for comparative analyses [38].

EXP is typically measured using RNA sequencing (RNA-Seq). Cell Model Passports processes the raw RNA-Seq data with RSEM [39] to obtain EXP values as raw read counts, FPKM, and TPM, while DepMap Public processes the raw data with Salmon [40] to obtain EXP values as $\log_2(\text{TPM}+1)$. Note that Cell Model Passports contains raw RNA-Seq data from both its own profiling and DepMap Public. Note that while Cell Model Passports states that EXP data are reported as $\log_2(\text{TPM}+1)$, the downloaded values appear to be untransformed TPMs. Only after manually applying a $\log_2(\text{TPM}+1)$-transformation do the values fall within a similar range as the DepMap Public data.

2.5.5 Proteomics

Proteomics refers to the large-scale study of proteins, including post-translational modifications, protein abundance, and protein-protein interaction (PPI). Since proteins carry out most biological functions within cells, proteomic data offer a closer reflection of the functional state of cells compared to transcriptomic data.

Protein abundance data are typically obtained using mass spectrometry-based methods, which are more technically challenging, time-consuming, and costly than transcriptomic approaches such as RNA-Seq for EXP. Cell Model Passports and DepMap Public provide a proteomics dataset, but with many missing values as some protein abundances were deemed unreliable due to high q-values. Cell Model Passports processes raw proteomics data with the MaxLFQ algorithm [41] to obtain protein intensity values per gene, while DepMap Public normalizes raw proteomics data using a pooled reference sample across batches [42] to obtain protein quantifications.

Post-translational modifications include protein folding (the acquisition of a functional three-dimensional structure, important for stability and activity) and, optionally, other modifications such as phosphorylation (the addition of phosphate groups, important for regulating protein activity and signaling) and glycosylation (the addition of sugar chains, important for stability, folding, and cell communication).

PPI networks model the physical and/or functional interactions between proteins within a cell. Since proteins often function in complexes or pathways, these networks provide contextual information about cellular processes that may not be captured by individual gene or protein features. PPI data are derived from experimental assays or computational predictions and are represented as graphs, with nodes denoting proteins (or the corresponding protein-coding genes) and edges denoting interactions. One example of a large database containing detailed protein links suitable for building PPI networks is STRING [43]. Figure 2.8 shows an example PPI network.

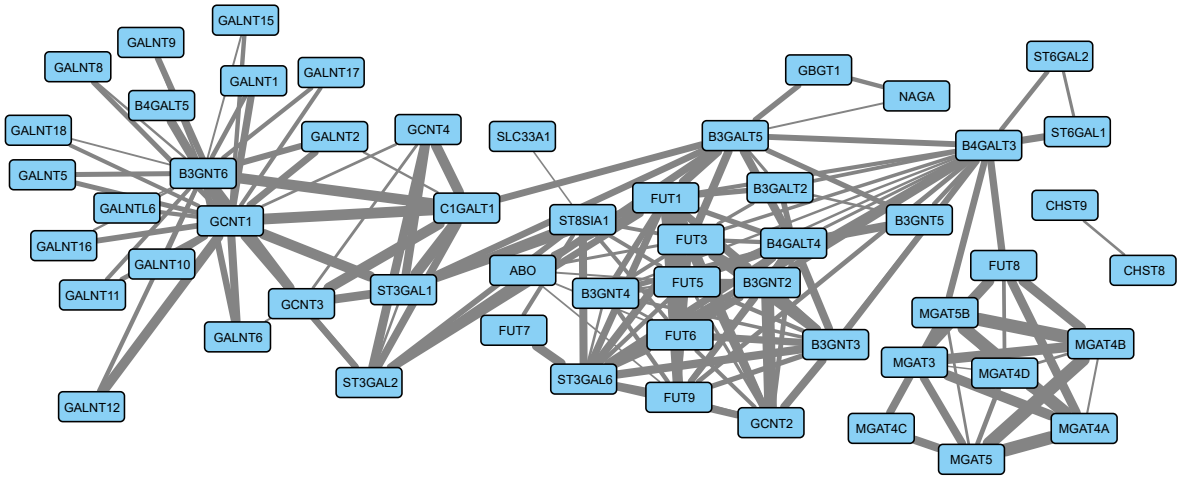


Figure 2.8: PPI network. A PPI network built from the STRING database is visualized in Cytoscape [44]. Each node represents a protein-coding gene, with edge width corresponding to the interaction confidence score from STRING. Only a subset of STRING genes is displayed, with edges below a confidence score of 0.9 excluded.

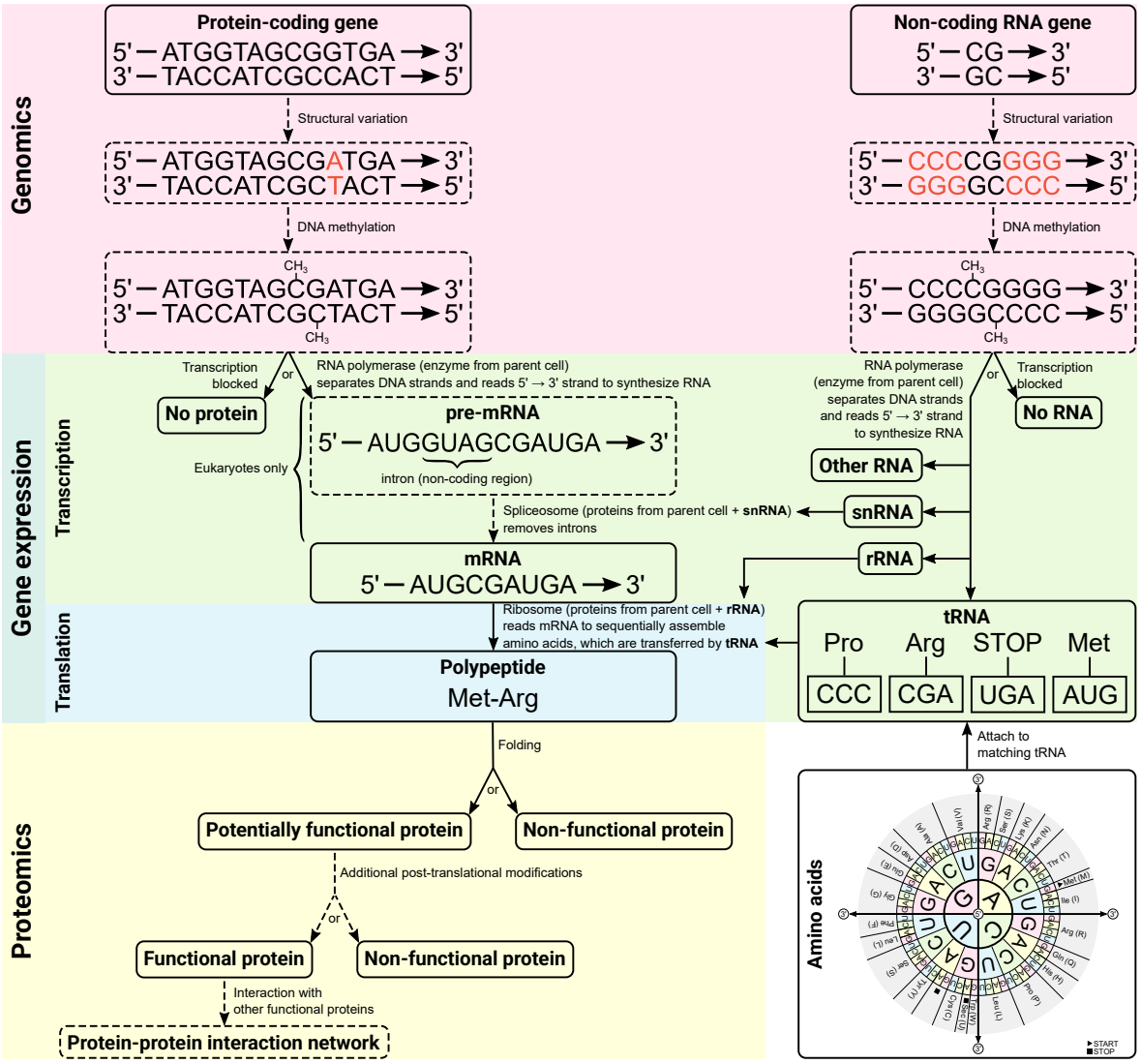


Figure 2.9: From gene to protein. On the right, a non-coding RNA gene with SVs may be silenced by DNA methylation or transcribed into snRNA, rRNA, tRNA, or other RNAs. Amino acids obtained from external sources (e.g., food) are attached to their cognate tRNAs. On the left, a protein-coding gene with an SV may be silenced by DNA methylation or undergo transcription, during which RNA polymerase separates the DNA strands and reads the 5' → 3' strand to synthesize mRNA directly (in prokaryotes) or pre-mRNA that, after splicing, becomes mature mRNA (in eukaryotes). The ribosome translates mRNA into a polypeptide by sequentially assembling amino acids delivered by tRNAs. Note that for illustration, the dipeptide Met-Arg is shown because a true polypeptide requires at least 10 amino acids. After translation, the polypeptide folds into either a non-functional or potentially functional protein. Additional post-translational modifications may further determine function. Functional proteins may interact with one another, forming the basis of PPI networks. Note that DNA methylation may promote rather than block transcription, and that prokaryotes have a single type of RNA polymerase for both coding and non-coding genes, whereas eukaryotes have several distinct RNA polymerases. The code wheel is included for visualization, indicating which tRNAs are charged with which amino acids. Dashed lines represent steps that may be bypassed.

2.6 Chemical input data

The following drug-derived, machine-readable data types are commonly considered as input features for current drug response prediction models. Figure 2.10 summarizes Sections 2.6.2 to 2.6.4.

2.6.1 Drug target data

Drug target data refer to the binding of proteins to ligands producing therapeutic effects. These targets are critical for understanding drug action and drug response.

Drug target data are typically acquired from experimental data and integrated databases that combine curated interactions and computational predictions. Examples of such resources are the STITCH [45] and HCDT 2.0 [46] databases, which aggregate known chemical-protein interactions from multiple sources. STITCH additionally includes predicted interactions; however, it has not been supported or updated since 2015.

2.6.2 Simplified molecular input line entry system

Simplified molecular input line entry system (SMILES) is a storage-efficient notation system that encodes the Lewis structure of a molecule as a string, capturing information about atoms, bonds, branching, ring structures, and aromaticity. Isomeric SMILES extend this format by additionally encoding stereochemistry and isotopes. Note that although technically imprecise, the term SMILES is commonly used both as a singular and plural noun, and often serves as shorthand for SMILES string(s).

Standard and isomeric SMILES can be retrieved from chemical databases such as PubChem [5] or ChEMBL [47]. Since a single molecular structure may be represented by multiple SMILES strings, both types can be generated following canonicalization rules to obtain canonical (isomeric) SMILES, ensuring a standardized representation.

2.6.3 Molecular graphs

Molecular graphs represent drugs as graphs, reflecting the Lewis structure and connectivity of a molecule. Nodes and edges correspond to atoms and bonds, respectively. Nodes can encode features such as atom type (e.g., carbon (C) or oxygen (O)) or formal charge, while edges can encode features such as bond type (single, aromatic, double, or triple).

Molecular graphs can be constructed from SMILES using cheminformatics libraries such as PubChemPy [48] or RDKit [49], which convert chemical data into graph objects.

2.6.4 Molecular fingerprints

Molecular fingerprints are fixed-length binary vector representations encoding the presence (1) or absence (0) of particular chemical substructures or features within a drug. Their simplicity allows for fast similarity comparisons between molecular structures.

Fingerprints can be computed from molecular graphs. Cheminformatics libraries such as RDKit [49] or DeepChem [50] provide tools to generate various fingerprint types, which differ in the chemical substructures or features they consider.

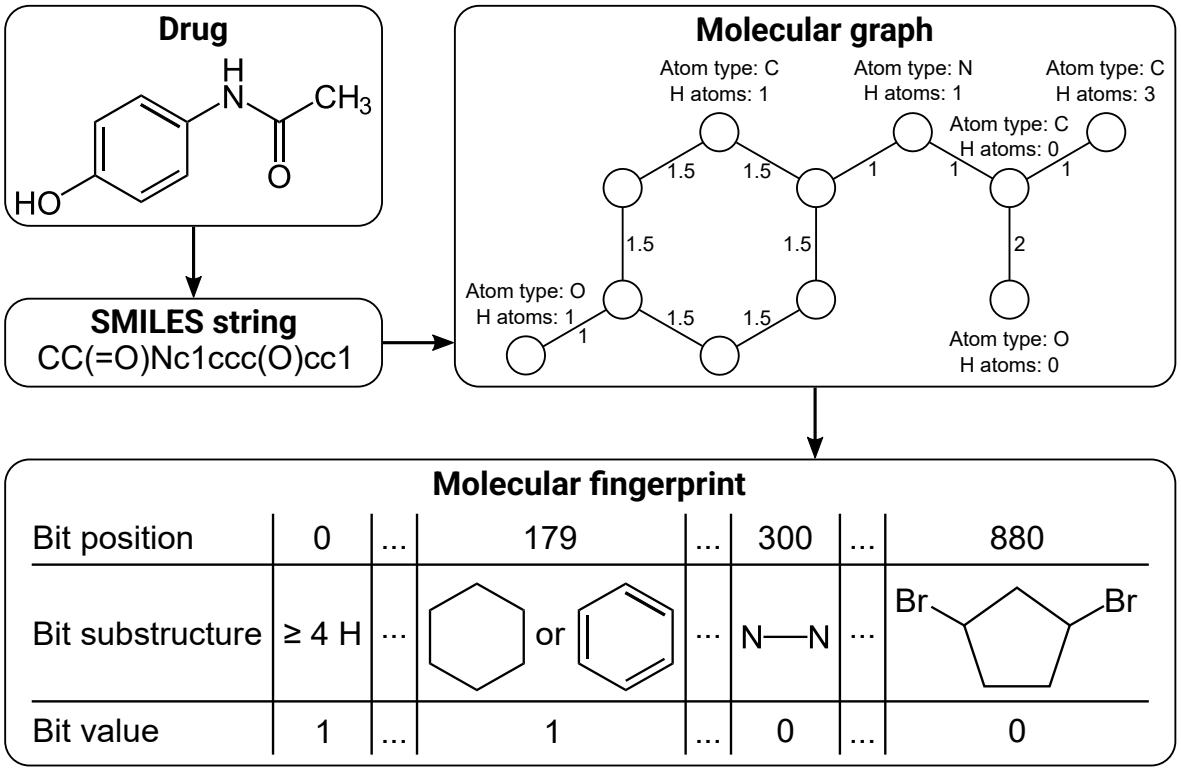


Figure 2.10: Machine-readable drug feature representations. A drug, here paracetamol, can be represented in machine-readable format as a SMILES string, a molecular graph, or a molecular fingerprint. The SMILES string is shown in canonical form. The molecular graph is constructed from the canonical SMILES and encodes node and edge features. In this example, the nodes capture the atom type and the number of implicitly connected hydrogen (H) atoms, while the edges represent bond types as weights. The molecular fingerprint is generated from the molecular graph. In this example, the PubChem fingerprint is illustrated, which encodes the presence (1) or absence (0) of 881 particular chemical substructures.

2.7 Graph neural networks

Graph neural networks (GNNs) are a class of neural networks (NNs) specifically designed to operate on data structured as graphs. Unlike traditional NNs, which operate on data such as images (grids) or sequences (texts), GNNs can model relationships and interactions in domains where data is best represented by graphs.

GNNs take a graph $G = (V, E)$ as input. Each node $u \in V$ has a feature vector $\mathbf{x}_u \in \mathbb{R}^{d_u}$, with $d_u \in \mathbb{N}$, and each edge $(u, v) \in E$ has a feature vector $\mathbf{e}_{(u,v)} \in \mathbb{R}^{d_e}$, with $d_e \in \mathbb{N}_0$. GNNs output embeddings at different levels, which can be used for various downstream tasks:

- node embeddings (e.g., for protein residues classification as binding or non-binding),
- edge embeddings (e.g., for PPI identification as activating, inhibiting, etc.),
- graph embedding (e.g., for molecular property prediction).

GNNs are built on a message-passing paradigm, where nodes iteratively update their representations by aggregating information from their neighbors and optionally from their incident edges, while edges update based on their incident nodes. See Figure 2.11 for an illustration of message passing. In the following, superscripts k are used to differentiate embeddings, parameters, and dimensions after k message-passing iterations. Particularly, $G^{(0)} = G$, $\mathbf{x}_u^{(0)} = \mathbf{x}_u$ and $\mathbf{e}_{(u,v)}^{(0)} = \mathbf{e}_{(u,v)}$. For each node u, the kth message-passing iteration to obtain the graph embedding $G^{(k)} = (\{\mathbf{x}_u^{(k)}\}_{u \in V}, \{\mathbf{e}_{(u,v)}^{(k)}\}_{(u,v) \in E})$ involves:

1. message aggregation:

$$\mathbf{m}_{\mathcal{N}(u)}^{(k-1)} = \text{AGGREGATE}^{(k-1)}\left(\left\{\text{MESSAGE}^{(k-1)}\left(\mathbf{x}_v^{(k-1)}, \mathbf{e}_{(u,v)}^{(k-1)}\right) \ \forall v \in \mathcal{N}(u)\right\}\right),$$

   where $\text{AGGREGATE}^{(k-1)}$ and $\text{MESSAGE}^{(k-1)}$ are arbitrary NNs, while $\mathcal{N}(u)$ denotes the set of neighbors of u,

2. node update to obtain the node embedding at layer k:

$$\mathbf{x}_u^{(k)} = \text{UPDATE}_u^{(k-1)}\left(\mathbf{x}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k-1)}\right),$$

   where $\text{UPDATE}_u^{(k-1)}$ is an arbitrary NN, and

3. edge updates to obtain the edge embeddings at layer k:

$$\mathbf{e}_{(u,v)}^{(k)} = \text{UPDATE}_{(u,v)}^{(k-1)}\left(\mathbf{x}_u^{(k-1)}, \mathbf{x}_v^{(k-1)}, \mathbf{e}_{(u,v)}^{(k-1)}\right) \ \forall v \in \mathcal{N}(u),$$

   where each $\text{UPDATE}_{(u,v)}^{(k-1)}$ is an arbitrary NN.

To give a concrete GNN example, the most basic GNN message passing (without edge updates) is defined as

$$\mathbf{m}_{\mathcal{N}(u)}^{(k-1)} = \sum_{v \in \mathcal{N}(u)} \mathbf{x}_v^{(k-1)},$$

$$\mathbf{x}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{x}_u^{(k-1)} + \mathbf{W}_{\text{neighbors}}^{(k)} \mathbf{m}_{\mathcal{N}(u)}^{(k-1)} + \mathbf{b}^{(k)}\right),$$

$$\mathbf{e}_{(u,v)}^{(k)} = \mathbf{e}_{(u,v)}^{(k-1)},$$

where $\mathbf{W}_{\text{self}}^{(k)}$, $\mathbf{W}_{\text{neighbors}}^{(k)} \in \mathbb{R}^{d^{(k)} \times d^{(k-1)}}$, with $d^{(k)}, d^{(k-1)} \in \mathbb{N}$, are trainable weight matrices, $\mathbf{b}^{(k)} \in \mathbb{R}^{d^{(k)}}$ is the trainable bias, and σ is a non-linear activation function (e.g., ReLU). The arbitrary NNs are chosen as:

$$\text{MESSAGE}^{(k-1)}\left(\mathbf{x}_v^{(k-1)}, \mathbf{e}_{(u,v)}^{(k-1)}\right) = \mathbf{x}_v^{(k-1)},$$

$$\text{AGGREGATE}^{(k-1)}\left(\left\{\mathbf{x}_v^{(k-1)} \ \forall v \in \mathcal{N}(u)\right\}\right) = \sum_{v \in \mathcal{N}(u)} \mathbf{x}_v^{(k-1)},$$

$$\text{UPDATE}_u^{(k-1)}\left(\mathbf{x}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k-1)}\right) = \sigma\left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{x}_u^{(k-1)} + \mathbf{W}_{\text{neighbors}}^{(k)} \mathbf{m}_{\mathcal{N}(u)}^{(k-1)} + \mathbf{b}^{(k)}\right),$$

$$\text{UPDATE}_{(u,v)}^{(k-1)}\left(\mathbf{x}_u^{(k-1)}, \mathbf{x}_v^{(k-1)}, \mathbf{e}_{(u,v)}^{(k-1)}\right) = \mathbf{e}_{(u,v)}^{(k-1)} \ \forall v \in \mathcal{N}(u).$$

Popular, more sophisticated GNN architectures are graph convolutional network (GCN), GraphSAGE, graph attention network (GAT), and graph isomorphism network (GIN).

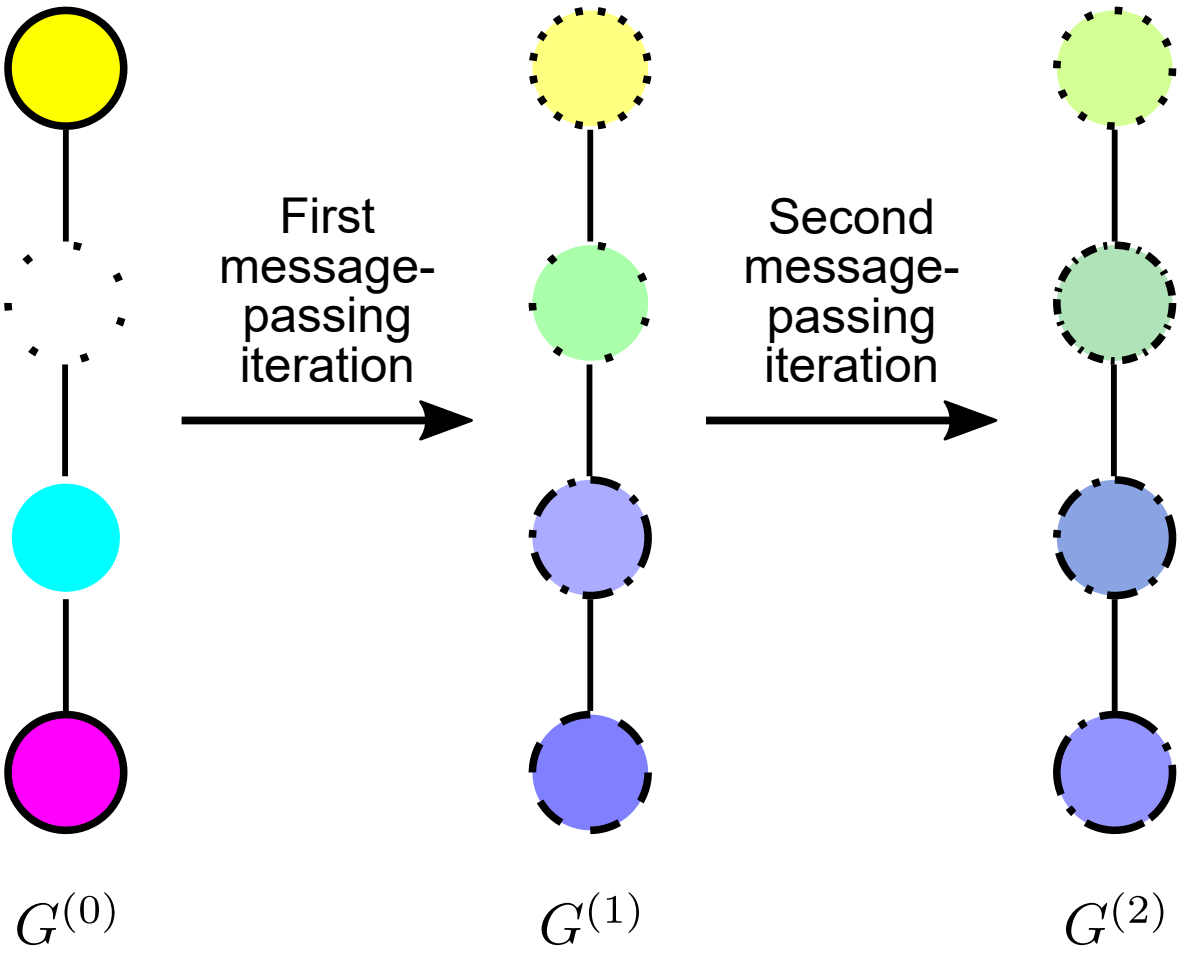


Figure 2.11: Two message-passing iterations. The nodes in the initial graph have two features: color and contour. At each iteration, node features are updated by aggregating their own features with those of their neighbors. For instance, after the first iteration, the third node from the top aggregates white, cyan, and magenta to produce a purple color, and aggregates dotted, absent, and solid contours to produce a dash-dot contour. Edge features are not updated. More precisely, in this example, $\text{MESSAGE}^{(k-1)}$ is an identity function (ignoring $\mathbf{e}_{(u,v)}^{(k-1)}$); $\text{AGGREGATE}^{(k-1)}$ sums the features of the neighbors; $\text{UPDATE}_{u}^{(k-1)}$ averages the features of u and its neighbors by adding the features of u to the aggregated sum and dividing by $|\mathcal{N}(u)|+1$; $\text{UPDATE}_{(u,v)}^{(k-1)}$ is an identity function (ignoring $\mathbf{x}_{u}^{(k-1)}$ and $\mathbf{x}_{v}^{(k-1)}$).

Chapter 3

Contributions

This chapter presents the three primary publications that form the core of this cumulative dissertation.

3.1 A critical review of multi-output support vector regression

This section provides a brief background and outlines the motivation behind the first publication. The publication itself is included thereafter.

3.1.1 Background and motivation

Common machine learning techniques include ensemble models such as random forests (RFs), kernel-based methods such as support vector regression (SVR), and deep learning architectures such as neural networks (NNs). Among these, black-box models, e.g., SVR and NNs, tend to outperform more interpretable models like RFs in terms of predictive performance [51]. Thus, the focus of this dissertation was laid on black-box models.

While NNs are well-suited for large datasets due to their linear scalability with sample size, SVR models tend to struggle with larger datasets because they scale quadratically [52]. However, SVR models are often more effective on small-sized datasets consisting of up to a few hundred samples [53]. Given that the DepMap datasets are of small size, the initial focus was on SVR.

However, unlike NNs, SVR has not been extensively explored in multi-output settings, and thus, only a limited number of multi-output SVR models exist. Because no comprehensive review was available, a systematic evaluation of all known multi-output SVR models to identify the most effective approach was conducted. Their performance was compared to that of single-output SVR and least-squares SVR models—yielding surprising results.

3.1.2 Publication

NK Tran and GW Klau wrote the manuscript; NK Tran designed and performed the research; NK Tran and LC Kühle developed the mathematical framework.

Pattern Recognition Letters 178 (2024) 69–75

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A critical review of multi-output support vector regression

Nguyen Khoa Tran *, Laura C. Kühle, Gunnar W. Klau

Heinrich Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, North Rhine-Westphalia, Germany

ARTICLE INFO

Editor: Bin Xiao

Dataset link: https://github.com/AlBi-HHU/Multi-Output_Workflow

Keywords:
Multi-output regression
Support vector
Least-squares

ABSTRACT

Single-output regression is a widely used statistical modeling method to predict an output based on one or more features of a datapoint. If a dataset has multiple outputs, they can be predicted independently from each other although this disregards potential correlations and thus may negatively affect the predictive performance. Therefore, multi-output regression methods predict multiple outputs simultaneously. One way to approach single-output regression is by using methods based on support vectors such as support vector regression (SVR) or least-squares SVR (LS-SVR). Based on these two, previous works have devised multi-output support vector regression methods. In this review, we introduce a unified notation to summarize the single-output support vector regression methods SVR and LS-SVR as well as state-of-the-art multi-output support vector regression methods. Furthermore, we implemented a workflow for subject- and record-wise bootstrapping and nested cross-validation experiments, which we used for an exhaustive evaluation of all single- and multi-output support vector regression methods on synthetic and non-synthetic datasets. Although the reviewed papers claim that their multi-output methods improve regression performance, we find that none of them outperform both single-output methods SVR and LS-SVR for various reasons. Due to these results, we reflected about the general concept of support vector regression and then concluded that support vector regression methods do not appear to be suitable for the task of multi-output regression.

1. Introduction

Given a dataset consisting of datapoints with one or more features and one output, single-output regression aims at accurately predicting the output for any datapoint by learning a mapping function from the feature(s) of a datapoint to the output. However, many datasets have multiple outputs, and although it is possible to treat the outputs separately, doing so disregards potential correlations among the outputs and thus does not make use of all the information available while learning each mapping function. Therefore, it might be beneficial to predict multiple outputs simultaneously, which is called multi-output regression (also known as multi-response, multi-variate, or multi-target regression).

Support vector regression (SVR), introduced by Drucker et al. [1], is an extension of the well-known support vector machines by Vapnik and Chervonenkis [2] and was devised for single-output regression. SVR uses a quadratic program (QP) to obtain the predictions of a single output. Another single-output support vector regression method is the least-squares SVR (LS-SVR) by Suykens and Vandewalle [3], which slightly changes the concept of SVR such that not a QP, but a system of linear equations derived from the QP needs to be solved in order to improve runtime. Although LS-SVR changes the concept of SVR, an empirical study by Van Gestel et al. [4] reports that the general performances of LS-SVR and SVR are similar. On this basis, previous works developed extensions for SVR or LS-SVR for the multi-output case. There are several applications for support vector regression and multi-output regression such as estimating different biophysical parameters from remote sensing images [5], real-time prediction for converter gas tank levels [6], or electric load forecasting [7]. Here, we review state-of-the-art multi-output support vector regression methods, namely multi-dimensional SVR (M-SVR) by Pérez-Cruz et al. [8], extended LS-SVR (ELS-SVR) by Zhang et al. [9], and multi-output LS-SVR (MLS-SVR) by Xu et al. [10]. Those three methods apply mathematical changes to the QP of SVR or LS-SVR such that outputs are not only predicted simultaneously, but also influenced by each other during the prediction process without the need to combine support vector regression with other techniques. More details and a unified notation for all support vector regression methods will be introduced in Section 2.

Contrary to previous claims, we do not find an improvement in the regression performance by using multi-output support vector regression when testing all methods more thoroughly than in the original papers. Our testing procedure involves bootstrapping as well as nested cross-validation (CV) experiments. First, we ran tests on own synthetic datasets to simulate cases where multi-output regression methods may

* Corresponding author.
E-mail addresses: nguyen.tran@hhu.de (N.K. Tran), laura.kuehle@hhu.de (L.C. Kühle), gunnar.klau@hhu.de (G.W. Klau).

https://doi.org/10.1016/j.patrec.2023.12.007
Received 26 June 2023; Received in revised form 9 November 2023; Accepted 18 December 2023
Available online 21 December 2023

have an advantage over single-output regression methods. However, none of the multi-output methods outperformed the two single-output methods. Second, we reproduced selected experiments from the original papers with bootstrapping. Surprisingly, our results do not seem fully congruent with those of the original authors. Hence, we investigated possible reasons for these two outcomes and discussed whether the concept of support vector regression is suitable for multi-output regression at all.

We came to the conclusion that at the moment, artificial neural networks should always be preferred over support vector regression for multi-output regression. In order to enable reproducing our results as well as testing new multi-output regression methods in a standardized way, we provide a workflow for benchmarking, which is publicly available on https://github.com/AlBi-HHU/Multi-Output_Workflow.

2. Methods

For all the following support vector regression methods, the input data consists of two matrices: a feature matrix $\boldsymbol{x} \in \mathbb{R}^{n\times d}$ and an output matrix $\boldsymbol{y} \in \mathbb{R}^{n\times p}$, where n is the number of datapoints, d the number of features, and p the number of outputs.

2.1. Notation

A matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ will be written as a bold letter, with $A_{i,j}$ as its element in row i and column j, $\boldsymbol{A}_i \in \mathbb{R}^{1\times p}$ as its ith row, and $\boldsymbol{A}_{:,j} \in \mathbb{R}^{n\times 1}$ as its jth column. Matrices of size 1×1 are treated as scalars and thus not written in bold.

$\boldsymbol{A}^T$, $\boldsymbol{A}_i^T$, or $\boldsymbol{A}_{:,j}^T$ denotes the transpose of $\boldsymbol{A}$, $\boldsymbol{A}_i$, or $\boldsymbol{A}_{:,j}$, respectively. Furthermore, we write the all-ones vector as $\boldsymbol{e}_n = (1, 1, \dots, 1)^T \in \mathbb{R}^{n\times 1}$, the zero matrix as $\boldsymbol{0}_{n,m} \in \mathbb{R}^{n\times m}$, and the identity matrix as $\boldsymbol{I}^n \in \mathbb{R}^{n\times n}$.

Iverson brackets are denoted by square brackets, the Euclidean norm of $\boldsymbol{a} \in \mathbb{R}^{n\times 1}$ is written as $\|\boldsymbol{a}\|_2$, and $\boldsymbol{A}\circ\boldsymbol{B}$ denotes the element-wise product for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times p}$. Moreover, if we add $\boldsymbol{A} + \boldsymbol{b}^T$ where $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$, we treat this as calculating $\boldsymbol{A} + \boldsymbol{B}$, where $\boldsymbol{B}$ consists of n copies of $\boldsymbol{b}$, i.e., $\boldsymbol{B} = (\boldsymbol{b}, \boldsymbol{b}, \dots, \boldsymbol{b})^T \in \mathbb{R}^{n\times p}$.

Lastly, let $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}, d < d'$, be a mapping function from one vector to another vector of higher dimension, which is needed for the kernel trick to enable non-linear support vector machines and regression. We extend the definition of ϕ for matrices $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and write $\phi(\boldsymbol{A})$ to map each row of $\boldsymbol{A}$ to a row of higher dimension, i.e., $\phi : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d'}$. For two matrices $\boldsymbol{x}_1 \in \mathbb{R}^{n\times d}$ and $\boldsymbol{x}_2 \in \mathbb{R}^{m\times d}$, the kernel function K for the kernel trick is defined as $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)\phi(\boldsymbol{x}_2)^T \in \mathbb{R}^{n\times m}$. Like all reviewed papers, we use the radial basis function (RBF) kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\gamma\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_2^2}$, $\gamma \in \mathbb{R}$ and $\gamma > 0$.

2.2. Support vector regression methods

The general concept behind all support vector regression methods is as follows: For each output j, we fit a hyperplane, which is described by a weight $\boldsymbol{w}_{:,j}$ and a bias b_j, in the feature space of $\boldsymbol{x}$ such that the sum of distances between output values $\boldsymbol{y}_{:,j}$ and hyperplane $\phi(\boldsymbol{x})\boldsymbol{w}_{:,j} + b_j$ is minimized. Any datapoint $\boldsymbol{x}^*$ in the same feature space can then have its jth output predicted by calculating $f(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)\boldsymbol{w}_{:,j} + b_j$. Since there may be multiple possible hyperplanes with the same sum of distances, the distances between a hyperplane and its two closest datapoints are maximized as a second criterion. We can control the importance of the second criterion by using a regularization parameter.

2.2.1. Support vector regression (SVR)

SVR surrounds the hyperplane with an ϵ-tube during fitting to allow for a prediction tolerance, see Fig. 1.

The primal QP is as follows:

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*}\ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\boldsymbol{e}_n^T(\boldsymbol{\xi}+\boldsymbol{\xi}^*)\\ \text{s. t. }\ & \phi(\boldsymbol{x})\boldsymbol{w} + b + \boldsymbol{\xi} \ge \boldsymbol{y} - \epsilon\\ & \phi(\boldsymbol{x})\boldsymbol{w} + b - \boldsymbol{\xi}^* \le \boldsymbol{y} + \epsilon\\ & \xi_i, \xi_i^* \ge 0 \qquad i = 1, \dots, n\end{aligned}$$

where $\boldsymbol{w} \in \mathbb{R}^{d'\times 1}$, $b \in \mathbb{R}$, $\boldsymbol{\xi}, \boldsymbol{\xi}^* \in \mathbb{R}^{n\times 1}$ are the primal variables (of which $\boldsymbol{\xi}$ as well as $\boldsymbol{\xi}^*$ are slack variables), and hyperparameters are regularization parameter $C > 0$ and prediction tolerance $\epsilon \ge 0$.

Usually, the dual QP, obtained from the Lagrangian function as well as Karush–Kuhn–Tucker (KKT) conditions, is used instead because then SVR can fit non-linear hyperplanes by using the kernel function:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}\ & g\\ \text{s. t. }\ & \boldsymbol{e}_n^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0\\ & 0 \le \alpha_i, \alpha_i^* \le C \qquad i = 1, \dots, n\end{aligned}$$

where $\boldsymbol{\alpha}, \boldsymbol{\alpha}^* \in \mathbb{R}^{n\times 1}$ are the dual variables and we define the objective function $g = \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T K(\boldsymbol{x}, \boldsymbol{x})(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon\boldsymbol{e}_n^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) - \boldsymbol{y}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)$.

After solving the QP, we compute b with the partial derivatives $\partial_{\boldsymbol{\alpha}} g = K(\boldsymbol{x}, \boldsymbol{x})(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon - \boldsymbol{y}$ and $\partial_{\boldsymbol{\alpha}^*} g = -K(\boldsymbol{x}, \boldsymbol{x})(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon + \boldsymbol{y}$ as follows: If any α_i or α_i^* satisfies $0 < \alpha_i, \alpha_i^* < C$, we set

$$b = \frac{-\sum_{i:0<\alpha_i<C} \partial_{\alpha_i} g + \sum_{i:0<\alpha_i^*<C} \partial_{\alpha_i^*} g}{|\{i \mid 0 < \alpha_i < C\}| + |\{i \mid 0 < \alpha_i^* < C\}|},$$

otherwise, we take b as the midpoint of the following range:

$$\min_{\{i|\alpha_i=0\vee\alpha_i^*=C\}} \{\partial_{\alpha_i} g, \partial_{\alpha_i^*} g\} \le b \le \max_{\{i|\alpha_i^*=0\vee\alpha_i=C\}} \{\partial_{\alpha_i} g, \partial_{\alpha_i^*} g\}.$$

Finally, we can make predictions for $\boldsymbol{x}^*$ with

$$f(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)\boldsymbol{w} + b = K(\boldsymbol{x}^*, \boldsymbol{x})(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + b.$$

2.2.2. Least-squares support vector regression (LS-SVR)

Instead of using an ϵ-tube for tolerating predictions close to the hyperplane, LS-SVR penalizes all predictions. However, by squaring the slack variables in the objective function from the primal QP of SVR, predictions are penalized much less the closer they are to the hyperplane, see Fig. 2. Also, squaring the slack variables simplifies the constraints, which enables transforming the QP into a system of linear equations, speeding up the solving process tremendously.

The QP is as follows:

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}}\ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}\\ \text{s. t. }\ & \phi(\boldsymbol{x})\boldsymbol{w} + b + \boldsymbol{\xi} = \boldsymbol{y}\end{aligned}$$

where $\boldsymbol{w} \in \mathbb{R}^{d'\times 1}$, $b \in \mathbb{R}$, $\boldsymbol{\xi} \in \mathbb{R}^{n\times 1}$ are the variables, and the only hyperparameter is regularization parameter C.

By using the Lagrangian function as well as KKT conditions, we obtain the following system of linear equations with variables b and $\boldsymbol{\alpha} \in \mathbb{R}^{n\times 1}$:

$$\begin{pmatrix} b \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0, & \boldsymbol{e}_n^T \\ \boldsymbol{e}_n, & K(\boldsymbol{x}, \boldsymbol{x}) + \frac{1}{C}\boldsymbol{I}^n \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \boldsymbol{y} \end{pmatrix}.$$

After solving the system of linear equations, we can make predictions for $\boldsymbol{x}^*$ with

$$f(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)\boldsymbol{w} + b = K(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{\alpha} + b.$$

2.2.3. Multi-dimensional support vector regression (M-SVR)

Training an independent SVR for p outputs of a multi-output dataset can be seen as having a p-dimensional hypercubic ϵ-tube. The idea of M-SVR is to treat all output values for datapoints lying outside the ϵ-tube equally by using a p-dimensional hyperspherical ϵ-tube instead. For example, if datapoint $\boldsymbol{x}_1$ has output values $y_{1,1}$ and $y_{1,2}$, for a hypercubic

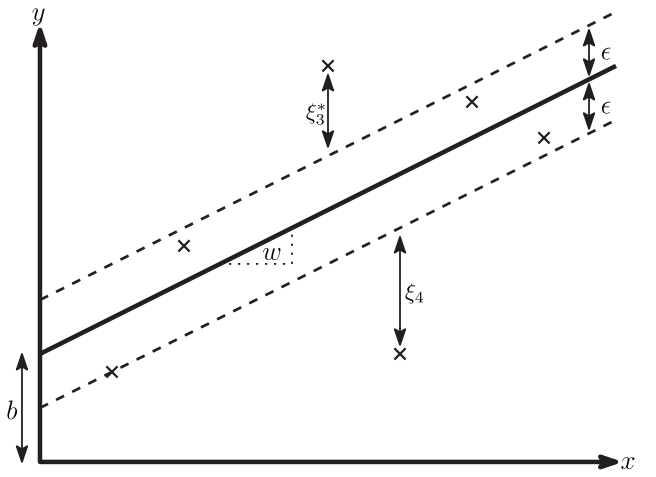


Fig. 1. An example of an SVR (linear hyperplane) with $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{6\times 1}$. A hyperplane (solid line) with slope w (dotted line) and intercept b is fitted to the datapoints. Datapoints outside the ϵ-tube (dashed lines) add ξ_i or ξ_i^* to the total penalty. Datapoints within the ϵ-tube are not penalized, i.e., $\xi_i = \xi_i^* = 0$.

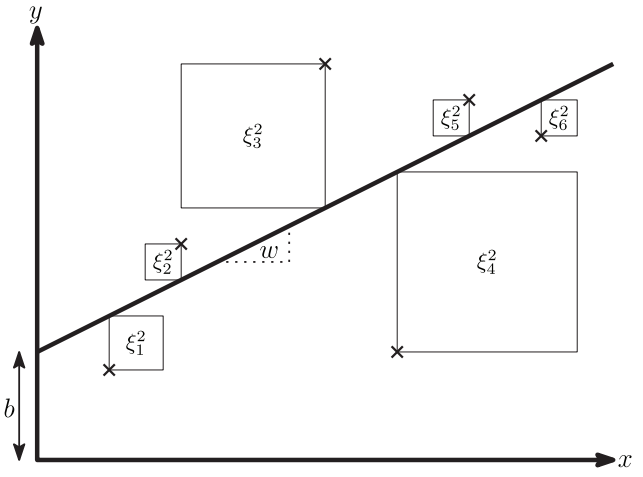


Fig. 2. An example of an LS-SVR (linear hyperplane) with $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{6\times 1}$. Similar to Fig. 1, a hyperplane (solid line) with slope w and intercept b is fitted to the datapoints. All datapoints add ξ_i^2 to the total penalty.

ϵ-tube we add one penalty if either $y_{1,1}$ or $y_{1,2}$ lies outside the ϵ-tube but two penalties if both lie outside, whereas for a hyperspherical ϵ-tube we add exactly one penalty in both cases, see Fig. 3.

Let $u_i = \|\boldsymbol{y}_i - \phi(\boldsymbol{x}_i)\boldsymbol{w} - \boldsymbol{b}^T\|_2$, then the QP is as follows:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\sum_{j=1}^{p} \boldsymbol{w}_{:,j}^T \boldsymbol{w}_{:,j} + C\boldsymbol{e}_n^T\boldsymbol{\xi}$$

$$\text{s. t. } u_i^2 - \epsilon \le \xi_i \qquad i = 1,\dots,n$$

$$\xi_i \ge 0 \qquad i = 1,\dots,n$$

where $\boldsymbol{w} \in \mathbb{R}^{d'\times p}$, $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$, $\boldsymbol{\xi} \in \mathbb{R}^{n\times 1}$ are the variables, and hyperparameters are regularization parameter $C > 0$ and prediction tolerance $\epsilon \ge 0$. Note that throughout this review, we set the value of the hyperspherical (red) ϵ to the value of the hypercubic (blue) ϵ multiplied by $2\cdot\pi^{-1/2}\cdot\Gamma(\frac{p}{2}+1)^{1/p}$ (Γ is the gamma function) such that the hyperspherical and hypercubic hypervolumes are equal.

The Lagrangian function that corresponds to the QP is $L_P(\boldsymbol{w},\boldsymbol{b}) = \frac{1}{2}\sum_{j=1}^{p}\boldsymbol{w}_{:,j}^T\boldsymbol{w}_{:,j} - C\sum_{i=1}^{n}[u_i^2 < \epsilon]\cdot(u_i^2-\epsilon)$. Because it is not differentiable at $u_i^2 = \epsilon$, Sánchez-Fernández et al. [11] instead propose to use $L_{P_2}(\boldsymbol{w},\boldsymbol{b}) = \frac{1}{2}\sum_{j=1}^{p}\boldsymbol{w}_{:,j}^T\boldsymbol{w}_{:,j} - C\sum_{i=1}^{n}[u_i < \epsilon]\cdot(u_i-\epsilon)^2$, which corresponds to the following QP:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\sum_{j=1}^{p} \boldsymbol{w}_{:,j}^T \boldsymbol{w}_{:,j} + C\boldsymbol{e}_n^T\boldsymbol{\xi}$$

$$\text{s. t. } (u_i - \epsilon)^2 \le \xi_i \qquad i = 1,\dots,n$$

$$\xi_i \ge 0 \qquad i = 1,\dots,n$$

We refer to this differentiable version as M-SVR2. In contrast to the QPs from the other support vector regression methods, the QP corresponding to L_P or L_{P_2} cannot be rewritten in a solvable dual QP by using the KKT conditions. Thus, Pérez-Cruz et al. [8] and Sánchez-Fernández et al. [11] use an iterative algorithm to approximate the result:

First, they treat each $a_i = [u_i^2 < \epsilon]\cdot C$ (M-SVR) or $a_i = [u_i < \epsilon]\cdot\frac{2C(u_i-\epsilon)}{u_i}$ (M-SVR2) as a constant to obtain $\boldsymbol{w}$ and $\boldsymbol{b}$ by solving the following system of linear equations with variables $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{n\times p}$:

$$\begin{pmatrix} \boldsymbol{b}^T \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_n^T\boldsymbol{a}, & \boldsymbol{a}^T K(\boldsymbol{x},\boldsymbol{x}) \\ \boldsymbol{e}_n, & K(\boldsymbol{x},\boldsymbol{x}) + \boldsymbol{D}^{-1} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{a}^T\boldsymbol{y} \\ \boldsymbol{y} \end{pmatrix}$$

where $\boldsymbol{w} = \phi(\boldsymbol{x})^T\boldsymbol{\alpha}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ with $D_{i,j} = [i = j]\cdot a_i$. Note that $\boldsymbol{D}$ is not invertible if $\exists i : a_i = 0$. Upon request, Sánchez-Fernández et al. [11] provided a way to solve the system of linear equations, which is to leave out the ith row and column from $K(\boldsymbol{x},\boldsymbol{x})$ and $\boldsymbol{D}$ as well as the ith row from $\boldsymbol{y}$ and $\boldsymbol{\alpha}$ for each $a_i = 0$ while setting α_i to 0.

Second, each a_i is recomputed from this solution. These two steps are repeated until L_P (M-SVR) or L_{P_2} (M-SVR2) converges. If all a_i are 0, we also stop.

Finally, we can make predictions for $\boldsymbol{x}^*$ with

$$f(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)\boldsymbol{w} + \boldsymbol{b}^T = K(\boldsymbol{x}^*,\boldsymbol{x})\boldsymbol{\alpha} + \boldsymbol{b}^T.$$

Note that $\boldsymbol{b}^T$ is missing in the original paper.

2.2.4. Extended least-squares support vector regression (ELS-SVR)

ELS-SVR models the multi-output regression problem as a single-output problem by flattening $\boldsymbol{y}$ and creating p copies of each datapoint $\boldsymbol{x}_i$ while keeping the original outputs by adding one-hot encodings as features. The new feature matrix $\boldsymbol{x}'$ and output matrix $\boldsymbol{y}'$ are defined as

$$\boldsymbol{x}' = \begin{pmatrix} \boldsymbol{I}_1^p, & \boldsymbol{x}_1 \\ \boldsymbol{I}_2^p, & \boldsymbol{x}_1 \\ \vdots, & \vdots \\ \boldsymbol{I}_p^p, & \boldsymbol{x}_1 \\ \boldsymbol{I}_1^p, & \boldsymbol{x}_2 \\ \vdots, & \vdots \\ \boldsymbol{I}_p^p, & \boldsymbol{x}_n \end{pmatrix} \in \mathbb{R}^{(n\cdot p)\times(p+d)}, \boldsymbol{y}' = \begin{pmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,p} \\ y_{2,1} \\ \vdots \\ y_{n,p} \end{pmatrix} \in \mathbb{R}^{(n\cdot p)\times 1}.$$

Zhang et al. [9] introduced a parameter $\zeta \in [0,1]$ which denotes the similarity between outputs, where 1 means total similarity. The role of ζ is to adjust the kernel function K: If two datapoints are originally from the same output, K remains unchanged, otherwise K is multiplied by ζ. Formally, the adjusted kernel function K^ζ is defined as $K^\zeta(\boldsymbol{x}_i', \boldsymbol{x}_j') = K^\zeta((\boldsymbol{I}_k^p, \boldsymbol{x}_i), (\boldsymbol{I}_\ell^p, \boldsymbol{x}_j)) = \zeta^{[k\neq\ell]}K(\boldsymbol{x}_i,\boldsymbol{x}_j)$. We extend the definition of K^ζ for matrices $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{B} \in \mathbb{R}^{m\times d}$ and write $K^\zeta(\boldsymbol{A},\boldsymbol{B}) \in \mathbb{R}^{n\times m}$, where an entry in row i and column j is $K^\zeta(\boldsymbol{A}_i,\boldsymbol{B}_j)$.

The QP is as follows:

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\sum_{j=1}^{p}\boldsymbol{\xi}_{:,j}^T\boldsymbol{\xi}_{:,j}$$

$$\text{s. t. } \phi((\boldsymbol{I}_j^p, \boldsymbol{x}_i))\boldsymbol{w} + b_j + \xi_{i,j} = y_{i,j} \qquad i = 1,\dots,n; j = 1,\dots,p$$

where $\boldsymbol{w} \in \mathbb{R}^{d'\times 1}$, $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$, $\boldsymbol{\xi} \in \mathbb{R}^{n\times p}$ are the variables, and hyperparameters are regularization parameter $C > 0$ and $\zeta \in [0,1]$.

Analogous to LS-SVR, by using the Lagrangian function as well as KKT conditions, we obtain the following system of linear equations with

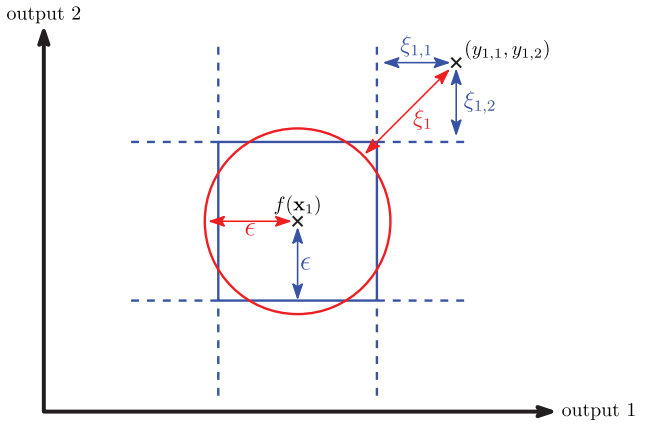


Fig. 3. An example of a hypercubic (blue) and a hyperspherical (red) ϵ-tube for a two-output regression. For each datapoint i, all p outputs are treated as p-dimensional coordinates $(y_{i,1}, y_{i,2}, \dots, y_{1,p})$. The red ϵ and blue ϵ are independent from each other, but here we set them such that the hyperspherical and hypercubic hypervolumes are equal. The prediction $f(\boldsymbol{x}_1)$ for datapoint $\boldsymbol{x}_1$ adds $\xi_{1,1}$ and $\xi_{1,2}$ (hypercubic) or ξ_1 (hyperspherical) to the total penalty.

variables $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{(n\cdot p)\times 1}$:

$$\begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{p,p}, & \boldsymbol{S}^T \\ \boldsymbol{S}, & K^{\zeta}(\boldsymbol{x}', \boldsymbol{x}') + \frac{1}{C}\boldsymbol{I}^{n\cdot p} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0}_{p,1} \\ \boldsymbol{y}' \end{pmatrix}$$

where $\boldsymbol{S} = (\boldsymbol{I}^p, \boldsymbol{I}^p, \dots, \boldsymbol{I}^p)^T \in \mathbb{R}^{(n\cdot p)\times p}$.

After solving the system of linear equations, we can make predictions for $\boldsymbol{x}^*$ with

$$\begin{aligned} f(\boldsymbol{x}^*)_{k,\ell} &= \phi((\boldsymbol{I}^p_{\ell}, \boldsymbol{x}^*_k))\boldsymbol{w} + b_{\ell} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{p} \alpha_{i,j} K^{\zeta}((\boldsymbol{I}^p_{\ell}, \boldsymbol{x}^*_k), (\boldsymbol{I}^p_j, \boldsymbol{x}_i)) + b_{\ell}. \end{aligned}$$

2.2.5. Multi-output least-squares support vector regression (MLS-SVR)

MLS-SVR is an extension of LS-SVR. Like for LS-SVR, we train one weight for each output, but we additionally train a weight $\boldsymbol{w_0}$ that is shared for the predictions of all outputs, see Fig. 4. The importance of $\boldsymbol{w_0}$ can be controlled by adjusting the regularization parameters of the other terms in the objective function.

The QP is as follows:

$$\min_{\boldsymbol{w_0}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}} \frac{1}{2}\boldsymbol{w_0}^T\boldsymbol{w_0} + \sum_{j=1}^{p} \frac{C'}{2p}\boldsymbol{w}_{:,j}^T\boldsymbol{w}_{:,j} + \frac{C}{2}\boldsymbol{\xi}_{:,j}^T\boldsymbol{\xi}_{:,j}$$

$$\text{s. t. } \phi(\boldsymbol{x})\boldsymbol{w_0} + \phi(\boldsymbol{x})\boldsymbol{w}_{:,j} + b_j + \boldsymbol{\xi}_{:,j} = \boldsymbol{y}_{:,j} \qquad j = 1, \dots, p$$

where $\boldsymbol{w_0} \in \mathbb{R}^{d'\times 1}$, $\boldsymbol{w} \in \mathbb{R}^{d'\times p}$, $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$, $\boldsymbol{\xi} \in \mathbb{R}^{n\times p}$ are the variables, and hyperparameters are two regularization parameters $C, C' > 0$.

Using the Lagrangian function as well as the KKT conditions analogously to LS-SVR would lead to a large system of linear equations due to many repetitions in the matrices, and therefore, Zhu and Gao [12] introduced a memory-efficient improvement. Broadly speaking, they avoid the repetitions by decomposing the large system into one system for each output and summing up these smaller systems such that the following system of linear equations is obtained with variables $\boldsymbol{b} \in \mathbb{R}^{p\times 1}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{n\times p}$:

$$\begin{pmatrix} \boldsymbol{b}^T \\ \boldsymbol{\alpha} \end{pmatrix} = \boldsymbol{B}^{-1}\boldsymbol{R} - \frac{1}{p}(\boldsymbol{B}^{-1} - \boldsymbol{G}^{-1})\boldsymbol{R}'$$

where $\boldsymbol{B} = \begin{pmatrix} 0, & \boldsymbol{e}_n^T \\ \boldsymbol{e}_n, & K(\boldsymbol{x}, \boldsymbol{x}) + \frac{1}{C}\boldsymbol{I}^n \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$, $\boldsymbol{R} = \begin{pmatrix} \mathbf{0}_{1,p} \\ \boldsymbol{y} \end{pmatrix} \in \mathbb{R}^{(n+1)\times p}$, $\boldsymbol{R}' = (\boldsymbol{R}\boldsymbol{e}_p, \boldsymbol{R}\boldsymbol{e}_p, \dots, \boldsymbol{R}\boldsymbol{e}_p) \in \mathbb{R}^{(n+1)\times p}$, and $\boldsymbol{G} = \boldsymbol{B} + \begin{pmatrix} 0, & \mathbf{0}_n^T \\ \mathbf{0}_n, & pK(\boldsymbol{x}, \boldsymbol{x}) \end{pmatrix} \in \mathbb{R}^{(n+1)\times p}$.

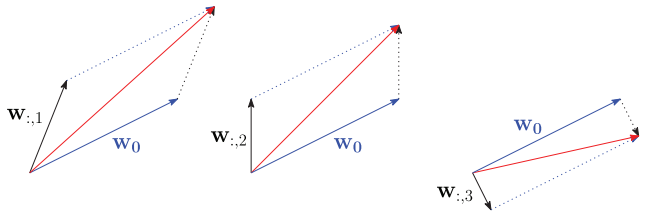


Fig. 4. An example for the intuition behind MLS-SVR for a three-output regression. $\boldsymbol{w_0}$ (blue) is the same for all three outputs, and we add $\boldsymbol{w}_{:,j}$, $j \in \{1, 2, 3\}$, to obtain the desired total weight (red) for each output.

Let $\boldsymbol{w_0} = (\boldsymbol{w_0}, \boldsymbol{w_0}, \dots, \boldsymbol{w_0}) \in \mathbb{R}^{d'\times p}$, and $\boldsymbol{\alpha}' = (\boldsymbol{\alpha}\boldsymbol{e}_p, \boldsymbol{\alpha}\boldsymbol{e}_p, \dots, \boldsymbol{\alpha}\boldsymbol{e}_p) \in \mathbb{R}^{n\times p}$. After solving the system of linear equations, we can make predictions for $\boldsymbol{x}^*$ with

$$\begin{aligned} f(\boldsymbol{x}^*) &= \phi(\boldsymbol{x}^*)\boldsymbol{w_0} + \frac{p}{C'}\phi(\boldsymbol{x}^*)\boldsymbol{w} + \boldsymbol{b}^T \\ &= K(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{\alpha}' + \frac{p}{C'}K(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{\alpha} + \boldsymbol{b}^T. \end{aligned}$$

3. Results

We implemented a workflow for benchmarking multi-output regression methods using Python 3 and the workflow management system Snakemake by Mölder et al. [13] For the implementation of SVR, we used the Scikit-learn package by Pedregosa et al. [14] for machine learning. The workflow can be used for bootstrapping or nested CV experiments with subject- or record-wise resampling for output matrix $\boldsymbol{y}$. See Fig. 5 for an illustration of subject- and record-wise bootstrapping as well as nested CV.

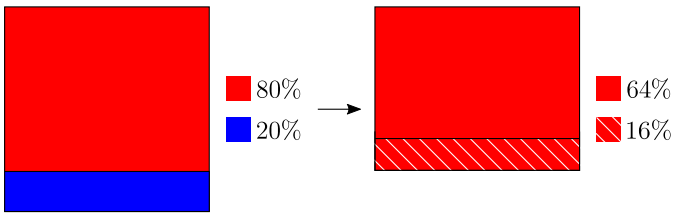
Bootstrapping describes any resampling method that uses random sampling with replacement. Here, we sample k percent of $\boldsymbol{y}$ to be in the test set for performance evaluation, while the remaining data, which we call outer training set, is used for hyperparameter tuning. For each sampling, we randomly choose k' percent from the outer training set to be in the validation set and refer to the data that is neither in the test nor in the validation set as inner training set. We repeat the sampling r times.

Similarly, nested CV is another resampling method with an inner k'-fold CV for hyperparameter tuning within an outer k-fold CV for performance evaluation. We decided to use nested CV instead of normal CV to ensure that the performance is evaluated solely on data that was unseen during training, and we also designed our bootstrapping accordingly.

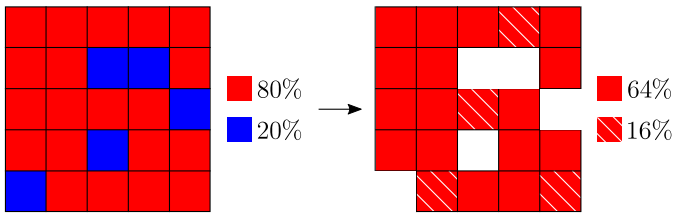
Subject-wise regression aims at predicting all outputs for new datapoints, whereas record-wise regression aims at predicting missing datapoint-output pairs. Although record-wise regression was not considered in the original papers, we included it in our analysis; we deemed it important not to exclude the possibility that the multi-output support vector regression methods perform well in this setting since it could be beneficial for datasets with multiple outputs and missing datapoint-output pairs. For example, the Genomics of Drug Sensitivity in Cancer project by Yang et al. [15] collects cancer cell data and their responses to different drugs, and approximately 15% of all cell-drug pairs are missing. An accurate imputation of missing values may facilitate data generation as we would no longer need to acquire all datapoint-output pairs.

3.1. Multi-output vs. single-output

First, we reflect on cases where multi-output regression could be advantageous. For the sake of completeness, we would like to mention that if all outputs are uncorrelated, multi-output regression cannot be advantageous in any case. If there are correlations between the outputs, we distinguish between two cases for subject-wise regression:



(a) Subject-wise resampling, outer split into outer training and test set (left) and inner split into inner training and validation set (right).



(b) Record-wise resampling, outer split into outer training and test set (left) and inner split into inner training and validation set (right).

Fig. 5. An illustration of splitting output matrix $\boldsymbol{y}$ into test, outer training, validation, and inner training set with (a) subject-wise or (b) record-wise resampling. Both examples show one sampling with $k = k' = 20\%$ (bootstrapping) or one outer and one inner iteration with $k = k' = 5$ (nested CV), where we take $1/5$ (= 20%) of the whole data as the test set (blue), $4/5$ (= 80%) as the outer training set, $1/5$ (= 16%) of the outer training set as the validation set (shaded) and $4/5$ (= 64%) as the inner training set. The outer 5-fold CV has five iterations where each datapoint will be part of the test set in exactly one of the iterations, and the inner 5-fold CV also has five iterations where each datapoint will be part of the validation set in exactly one of the iterations. All hyperparameter combinations are used for training on the inner training set, and after training, the performance of each combination is measured on the validation set, i.e., for nested CV, each combination has five performances (one in each iteration). To complete the hyperparameter tuning, we choose the hyperparameter combination with the best performance (bootstrapping) or best average performance calculated from the five performances (nested CV) in order to train on the outer training set. Then, we measure the performance on the test set. To complete the performance evaluation, after all r repetitions (bootstrapping) or five outer iterations (nested CV) we could for example report the median performance.

- If there are neither missing values nor outliers in $\boldsymbol{y}$, those correlations between outputs where we can find a (reasonably simple) mapping from one output to the other should not be able to improve regression performance much, if at all: In the best case we would have a perfect correlation, e.g., if we have two outputs with a perfect linear correlation, mapping one output to the other means we have two identical outputs and thus not an information gain compared to only one output.
- If there are missing values or outliers in $\boldsymbol{y}$, they should have less negative influence on multi-output than on single-output methods when learning the mapping function from feature(s) to outputs. For example, if we have two outputs that are identical except for a missing value as well as an outlier in output 1, both would affect a single-output method more (when training on output 1) than they would affect a multi-output method because the latter has about twice as much information per datapoint during training to shift the regression to the correct direction.

For record-wise regression, the detection of correlations between the outputs could benefit multi-output methods in the same way as it does for subject-wise regression, and additionally when predicting the missing values that occur due to record-wise resampling. For example, if there are two outputs with a strong positive correlation and only one missing datapoint-output pair while all other outputs of the same datapoint have a high value, the missing pair is also likely a high value.

3.2. Experiments

We ran our bootstrapping and nested CV experiments on three synthetic datasets to check whether the multi-output support vector regression methods have any of the advantages described in Section 3.1.

- The first dataset is to check for the advantage when missing values occur due to record-wise resampling.
- The second dataset is for checking whether the multi-output methods can use the correlation between the outputs with no simple mapping between the outputs to their advantage when testing subject-wise or record-wise.
- The third dataset checks for the advantage when outliers appear in the dataset when testing subject-wise or record-wise.

Note that contrary to the single-output methods SVR and LS-SVR where missing values can simply be ignored by removing the corresponding rows from $\boldsymbol{x}$ and $\boldsymbol{y}$, all three multi-output support vector regression methods build a large system of linear equations where missing values can be neither ignored nor removed. Hence, for the record-wise experiments with all three multi-output methods, we chose to impute each missing value in output j with the corresponding column mean, i.e., the mean of all non-missing values in $\boldsymbol{y}_{:,j}$. We ignored the case of missing values in $\boldsymbol{y}$ (those that do not occur due to record-wise resampling) because the mean imputation makes it similar to the case of outliers occurring in $\boldsymbol{y}$.

We generated a feature matrix $\boldsymbol{x} \in \mathbb{R}^{1000\times 1}$ by sampling 1000 times from a uniform distribution $U[0, 1)$ and sorting the values in ascending order. This feature matrix is part of all three synthetic datasets. We then generated the following six outputs $\in \mathbb{R}^{1000\times 1}$:

- $\boldsymbol{y_1} = \boldsymbol{x} \circ \boldsymbol{x}$ (each entry in $\boldsymbol{x}$ is squared),
- $\boldsymbol{y_2} = 0.5\boldsymbol{y_1}$,
- $\boldsymbol{y_3}$ is generated by sampling 1000 times from a normal distribution $\mathcal{N}(0, 1)$ and sorting in ascending order,
- $\boldsymbol{y_4}$ is generated by sampling 1000 times from a uniform distribution $U[0, 1)$ and sorting in ascending order,
- $\boldsymbol{y}_1^{\text{outliers}}$ is $\boldsymbol{y_1}$, but we increase every 50th value by 10,
- $\boldsymbol{y}_2^{\text{outliers}}$ is $\boldsymbol{y_2}$, but we increase every 100th value by 5.

Finally, the three output matrices of our synthetic datasets are $\boldsymbol{y_{1,2}} = (\boldsymbol{y_1}, \boldsymbol{y_2}) \in \mathbb{R}^{1000\times 2}$, $\boldsymbol{y_{3,4}} = (\boldsymbol{y_3}, \boldsymbol{y_4}) \in \mathbb{R}^{1000\times 2}$, and $\boldsymbol{y}_{1,2}^{\text{outliers}} = (\boldsymbol{y}_1^{\text{outliers}}, \boldsymbol{y}_2^{\text{outliers}}) \in \mathbb{R}^{1000\times 2}$.

We measure the predictive performance with the mean absolute error $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f(x_i^*) - y_i|$ or root-mean-square error $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i^*) - y_i)^2}$. For the experiments on our synthetic datasets, we set $k = k' = 50\%$ and $r = 10$ for bootstrapping and $k = k' = 5$ for the nested CV. We considered the following values for the hyperparameter tuning: $C \in \{0.1, 1, 10, 100\}$, $\epsilon \in \{10^1, 10^{-1}, 10^{-3}\}$, $C' \in \{0.1, 1, 10\}$ (only for MLS-SVR), and $\zeta \in \{0.25, 0.5, 0.75, 1\}$ (only for ELS-SVR). We set the kernel parameter $\gamma = \frac{1}{d \cdot var(\boldsymbol{x})}$, where $var(\boldsymbol{x})$ is the variance of $\boldsymbol{x}$ and d is the number of features in $\boldsymbol{x} \in \mathbb{R}^{n\times d}$.

We now analyze the results; the corresponding plots are given in the Supplementary Material (Section 1). In all record-wise experiments on all three datasets $\boldsymbol{y_{1,2}}$, $\boldsymbol{y_{3,4}}$, and $\boldsymbol{y}_{1,2}^{\text{outliers}}$, the single-output method SVR performs best. This outcome is not unexpected though because all three multi-output support vector regression methods need to impute their missing values, and as mentioned above, the multi-output methods were not developed for this setting. Strikingly, in all subject-wise experiments on $\boldsymbol{y_{3,4}}$ and $\boldsymbol{y}_{1,2}^{\text{outliers}}$, SVR comes out on top again, which would mean that there exists no single case where the three multi-output support vector regression methods have an advantage over SVR.

3.3. Reproducing original experiments

As the authors of the multi-output support vector regression methods [8–10] claim to have achieved good results, we decided to reproduce their experiments on their own synthetic datasets with the same hyperparameters in order to verify whether we overlooked cases in

which their methods can outperform single-output methods. Broadly speaking, the synthetic dataset of each paper uses a feature matrix $\boldsymbol{x}$ that consists of random values and an output matrix $\boldsymbol{y}$ that is obtained by applying trigonometric and related functions on $\boldsymbol{x}$ with added noise. A description for generating each dataset can be found in the Supplementary Material (Section 2).

For M-SVR the number of datapoints n as well as the data splitting procedure are not given, and for ELS-SVR ($n = 100$) and MLS-SVR ($n = 1000$) the data was split just once into 50% training and 50% test set (without validation set). Hence, we instead applied our subject-wise bootstrapping with $k = k' = 50\%$ with $r = 10$, and set $n = 1000$ in order to have the same n for all synthetic datasets.

Overall, the multi-output support vector regression methods are not superior to SVR or LS-SVR. The result plots are given in the Supplementary Material (Section 2).

4. Discussion

None of the multi-output support vector regression methods achieved good results in our experiments. The similar results of SVR and M-SVR2 indicate that the shape of the ϵ-tube does not have a significant impact on performance, but rather its hypervolume. We think that whether a hypercube or a hypersphere yields better results is dataset-dependent. The only case where we believe M-SVR could perform better than SVR is when all outliers co-occur in all outputs of the same datapoint, i.e., M-SVR would only add one penalty overall while SVR would add one penalty for each output, which might force the hyperplane to adjust to the outliers. However, it might be even more reasonable to exclude such datapoints instead of using M-SVR or M-SVR2.

Zhang et al. [9] claim that ELS-SVR outperformed SVR with hyperparameters $C = 100$, $\gamma = 0.1$, and $\zeta = 0.1$. We cannot confirm this because the median and mean MAEs and RMSEs are lower for SVR than for ELS-SVR, see Supplementary Figure 12. For SVR the standard deviation of the MAEs and RMSEs is much higher though, which might explain the results [9] obtained since they ran their experiment only once. Nevertheless, we detected some weaknesses of ELS-SVR, which show that the idea behind the similarity between outputs ζ seems reasonable in theory but not in practice: First, the method consumes $\mathcal{O}(n^2 \cdot p^2)$ memory for the kernel matrix $K^{\zeta}(\boldsymbol{x}', \boldsymbol{x}')$, which can quickly become infeasible for higher n and p. Second, we believe that for more than two outputs the similarity between outputs ζ needs to be chosen individually for each pair of outputs to be more effective. This would require a lot of additional computational power and still not capture negative correlations: If ζ were negative, Mercer's theorem would not be fulfilled, i.e., K^{ζ} would not be positive semi-definite so that the problem would become non-convex.

MLS-SVR actually performed marginally better than LS-SVR, see Supplementary Figure 13. Nonetheless, we also identified one major weakness of MLS-SVR: If two outputs j and k are strongly positively correlated, but a third output ℓ is less strongly correlated, uncorrelated, or even negatively correlated, a problem occurs for $\boldsymbol{w_0}$ which is intended as a common weight for the predictions of all outputs. We tested this by adding the uncorrelated, randomly created output $\boldsymbol{y_4}$ from Section 3.2 as a third output to the synthetic data of MLS-SVR, which led to LS-SVR marginally outperforming MLS-SVR, see Supplementary Figure 14.

On closer inspection, we believe that for multi-output regression any support vector regression method would be unsuited: In a QP with an objective function and constraints, to capture linear correlations between all pairs of outputs for example, we see no other option than adding a quadratic amount of constraints. However, for a better model we would need to capture many non-linear correlations as well, leading to an excessive amount of constraints and hence an infeasible runtime for building the QP.

Nevertheless, this does not mean multi-output regression is unusable: For example, we trained artificial neural networks (ANNs) on either each output independently or all outputs simultaneously with subject-wise resampling. Although we chose a very simple architecture for the ANNs, the multi-output approach already performed better than the single-output approach, and thus we believe a more sophisticated architecture would yield even better results. See the Supplementary Material (Section 3) for details on the ANN architectures and the result plots.

5. Conclusions

In this review, we summarized single- and multi-output support vector regression methods with unified notation and implemented a workflow for subject- and record-wise bootstrapping as well as nested CV experiments to test multi-output regression methods in a standardized way. Unfortunately, none of the multi-output support vector regression methods reliably outperformed both single-output support vector regression methods. We discussed possible reasons for this: the hyperspherical ϵ-tube of M-SVR does not exhibit an advantage over a hypercubic one, the similarity between outputs ζ of ELS-SVR cannot include negative correlations, and MLS-SVR does not appropriately handle the case of only partial correlations between outputs. All in all, we conclude that for multi-output regression multi-output ANNs should be preferred and multi-output support vector regression methods do not need further investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data and code in the manuscript (https://github.com/AlBi-HHU/Multi-Output_Workflow).

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. We thank Max J. Ried as well as the Center for Information and Media Technology at Heinrich Heine University Düsseldorf for their provided computational infrastructure and support.

Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2023.12.007.

References

[1] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, Vol. 9, MIT Press, 1996.
[2] V. Vapnik, A. Chervonenkis, Theory of pattern recognition, 1974, Nauka, Moscow.
[3] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.
[4] T. Van Gestel, J. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, J. Vandewalle, Benchmarking least squares support vector machine classifiers, Mach. Learn. 54 (2004) 5–32.
[5] D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, G. Camps-Valls, Multioutput support vector regression for remote sensing biophysical parameter estimation, IEEE Geosci. Remote Sens. Lett. 8 (4) (2011) 804–808.
[6] Z. Han, Y. Liu, J. Zhao, W. Wang, Real time prediction for converter gas tank levels based on multi-output least square support vector regressor, Control Eng. Pract. 20 (12) (2012) 1400–1409.
[7] Z. Zhang, W.-C. Hong, Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads, Knowl.-Based Syst. 228 (2021).

[8] F. Pérez-Cruz, G. Camps-Valls, E. Soria-Olivas, J. Pérez-Ruixo, A. Figueiras-Vidal, A. Artés-Rodríguez, Multi-dimensional function approximation and regression estimation, in: Artificial Neural Networks – ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings, Vol. 12, Springer, 2002, pp. 757–762.

[9] W. Zhang, X. Liu, Y. Ding, D. Shi, Multi-output LS-SVR machine in extended feature space, in: 2012 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings, IEEE, 2012, pp. 130–134.

[10] S. Xu, X. An, X. Qiao, L. Zhu, L. Li, Multi-output least-squares support vector regression machines, Pattern Recognit. Lett. 34 (9) (2013) 1078–1084.

[11] M. Sánchez-Fernández, M. de Prado-Cumplido, J. Arenas-García, F. Pérez-Cruz, SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems, IEEE Trans. Signal Process. 52 (8) (2004) 2298–2307.

[12] X. Zhu, Z. Gao, An efficient gradient-based model selection algorithm for multi-output least-squares support vector regression machines, Pattern Recognit. Lett. 111 (2018) 16–22.

[13] F. Mölder, K. Jablonski, B. Letcher, M. Hall, C. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. Twardziok, A. Kanitz, et al., Sustainable data analysis with Snakemake, F1000Research 10 (2021).

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[15] W. Yang, J. Soares, P. Greninger, E. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. Smith, I. Thompson, S. Ramaswamy, P. Futreal, D. Haber, M. Stratton, C. Benes, U. McDermott, M. Garnett, Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells, Nucleic Acids Res. (ISSN: 0305-1048) 41 (D1) (2012) D955–D961.

Supplementary Material for: A Critical Review of Multi-Output Support Vector Regression

Nguyen Khoa Tran[a,*], Laura C. Kühle[a] and Gunnar W. Klau[a]

[a]Heinrich Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, North Rhine-Westphalia, Germany

1. Results on our Synthetic Datasets

See pages 3 to 6 for the plots. In all record-wise experiments, the single-output method SVR performs best, see Figures 1 to 6. In all subject-wise experiments, again SVR comes out on top, see Figures 7 to 10.

2. Reproducing Original Experiments

See pages 7 and 8 for the plots.

For each of the three multi-output support vector regression methods, we describe the synthetic dataset generation, summarize the results of reproducing the original experiments, and discuss possible reasons for the outcome. For the description of generating the synthetic datasets, we extend the definition of any function $f : \mathbb{R} \to \mathbb{R}$ for matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and write $f(\boldsymbol{A})$ to apply f on each entry of $\boldsymbol{A}$.

2.1. Multi-Dimensional Support Vector Regression (M-SVR)

The synthetic dataset consists of feature matrix $\boldsymbol{x} \in \mathbb{R}^{n\times 2}$ and output matrix $\boldsymbol{y} \in \mathbb{R}^{n\times 5}$. Each feature column of $\boldsymbol{x}$ can be generated by sampling n times from a normal distribution $\mathcal{N}(0, 1)$. The output columns in $\boldsymbol{y}$ are:

1. $\boldsymbol{y}_{:,1} = 4\sin(\boldsymbol{x}_{:,1}) - 2\text{sinc}(\boldsymbol{x}_{:,2}) + 5 + \boldsymbol{n}_{:,1}$,
2. $\boldsymbol{y}_{:,2} = 3\sin(\boldsymbol{x}_{:,1}) - 3\cos(\boldsymbol{x}_{:,2}) + 2 + \boldsymbol{n}_{:,2}$,
3. $\boldsymbol{y}_{:,3} = -5\text{sinc}(\boldsymbol{x}_{:,1}) + 4\sin(\boldsymbol{x}_{:,2}) + 1 + \boldsymbol{n}_{:,3}$,
4. $\boldsymbol{y}_{:,4} = -2\sin(\boldsymbol{x}_{:,1}) - 4\sin(\boldsymbol{x}_{:,2}) - 5 + \boldsymbol{n}_{:,4}$,
5. $\boldsymbol{y}_{:,5} = 4\text{sinc}(\boldsymbol{x}_{:,1}) - 2\cos(\boldsymbol{x}_{:,2}) - 3 + \boldsymbol{n}_{:,5}$

where $\boldsymbol{n} \in \mathbb{R}^{n\times 5}$ is a matrix filled with values sampled from a normal distribution $\mathcal{N}(0, 0.5)$.

Hyperparameters are $C = 10$, $\gamma = 0.5$, and hypercube-$\epsilon = 1.5$ (i.e., blue ϵ in Figure 3). The original experiments were run for multiple ϵ values, where M-SVR outperformed SVR clearly for the hypersphere-ϵ (i.e., red ϵ in Figure 3) between 1 and 1.5, which is why we set hypercube-$\epsilon = 1.5$ as it falls within this range.

However, neither M-SVR nor the improved M-SVR2 outperforms SVR, see Figure 11.

2.2. Extended Least-Squares Support Vector Regression (ELS-SVR)

The synthetic dataset consists of feature matrix $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ and output matrix $\boldsymbol{y} \in \mathbb{R}^{n\times 5}$. The feature column of $\boldsymbol{x}$ can be generated by sampling n times from a uniform distribution $U(0, 1)$. The output columns in $\boldsymbol{y}$ are:

1. $\boldsymbol{y}_{:,1} = \boldsymbol{x}\cos(2\pi\boldsymbol{x}) + \boldsymbol{n}_{:,1}$,
2. $\boldsymbol{y}_{:,2} = \boldsymbol{x}\sin(2\pi\boldsymbol{x}) + \boldsymbol{n}_{:,2}$

where $\boldsymbol{n} \in \mathbb{R}^{n\times 2}$ is a matrix filled with values sampled from a normal distribution $\mathcal{N}(0, 0.05)$.

ELS-SVR does not outperform SVR with hyperparameters $C = 100$, $\gamma = 0.1$, and $\zeta = 0.1$, see Figure 12.

2.3. Multi-Output Least-Squares Support Vector Regression (MLS-SVR)

The synthetic dataset consists of a simulated time series process with feature matrix $\boldsymbol{x} \in \mathbb{R}^{n\times 4}$ and output matrix $\boldsymbol{y} \in \mathbb{R}^{n\times 2}$. Each row i of $\boldsymbol{x}$ is $\boldsymbol{x}_i = (\boldsymbol{y}_{i-1,1}, \boldsymbol{y}_{i-2,1}, \boldsymbol{y}_{i-1,2}, \boldsymbol{y}_{i-2,2})$. The two outputs at time i in $\boldsymbol{y}$ are:

1. $\boldsymbol{y}_{i,1} = 0.1\sin(\pi\boldsymbol{y}_{i-1,2}) + (0.8 - 0.5\exp(-\boldsymbol{y}_{i-1,1}^2))\boldsymbol{y}_{i-1,1} - (0.3 + 0.9\exp(-\boldsymbol{y}_{i-1,1}^2))\boldsymbol{y}_{i-2,1} + \boldsymbol{n}_{i,1}$,
2. $\boldsymbol{y}_{i,2} = 0.6\boldsymbol{y}_{i-1,2} + 0.2\boldsymbol{y}_{i-1,2}\boldsymbol{y}_{i-2,2} + 1.2\tanh(\boldsymbol{y}_{i-2,1}) + \boldsymbol{n}_{i,2}$

where $y_{0,1} = y_{-1,1} = y_{0,2} = y_{-1,2} = 0$ and the zero-mean Gaussian noise $\boldsymbol{n} \in \mathbb{R}^{n\times 2}$ has a covariance $\sigma\boldsymbol{I}^2$. **?** use $\sigma \in \{0.01, 0.02, 0.03, 0.04\}$.

The hyperparameters considered for hyperparameter tuning are $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$, $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$, and $C' \in \{2^{-10}, 2^{-8}, \ldots, 2^{10}\}$.

In some cases, MLS-SVR actually performed marginally better than LS-SVR, see Figure 13, but not in all, see Figure 14.

3. Results for Artificial Neural Networks

See page 9 for the plots.

Our ANNs start with an input layer of dimension d, followed by b blocks, and end with the output layer of dimension p. A block consists of the following three components: fully-connected layer of dimension h, activation function, dropout layer. The ANNs consider the following hyperparameters for hyperparameter tuning: $b \in \{2, 3, 4\}$, $h \in \{64, 256, 1024\}$, and learning rate $\in \{0.001, 0.01, 0.1\}$. The other hyperparameters were fixed: activation function = ReLU, dropout probability = 0.5, optimizer = stochastic gradient descent with 0.95 momentum, batch size = 32, number of optimizer steps per epochs = 10, threshold to determine after how many epochs without improvement in loss to stop = 10.

*Corresponding author

nguyen.tran@hhu.de (N.K. Tran); laura.kuehle@hhu.de (L.C. Kühle); gunnar.klau@hhu.de (G.W. Klau)

ORCID(s): 0000-0002-4732-4294 (N.K. Tran); 0009-0000-3581-0677 (L.C. Kühle); 0000-0002-6340-0090 (G.W. Klau)

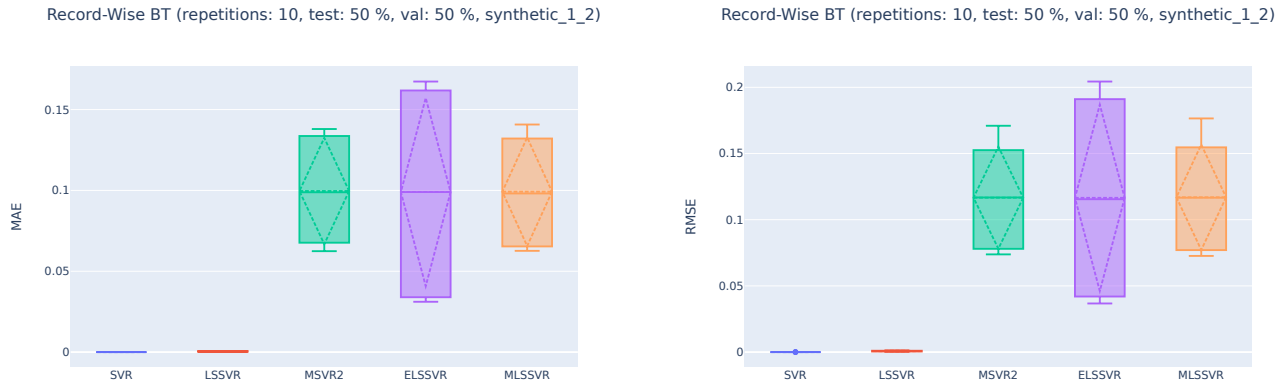


Figure 1: Results of record-wise bootstrapping with $k = k' = 50\,\%$ and $r = 10$ for the synthetic dataset $\boldsymbol{y}_{1,2}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 20 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

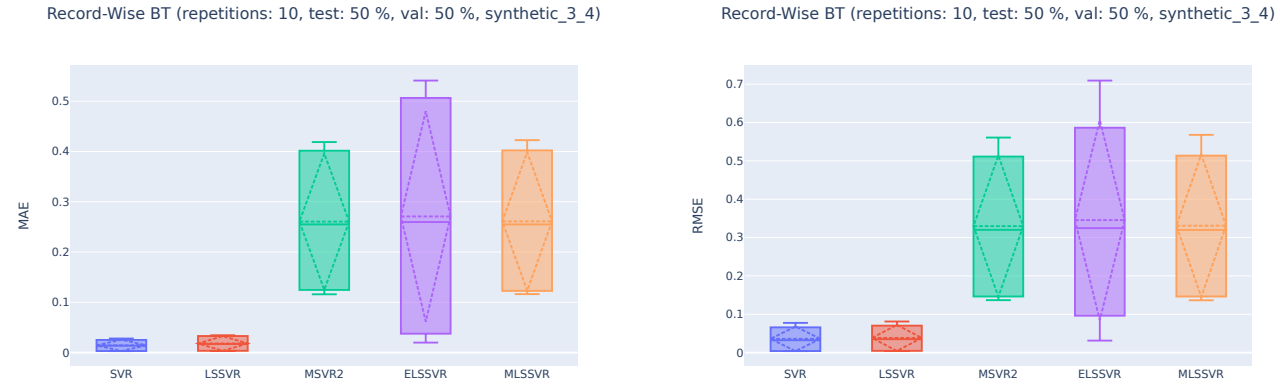


Figure 2: Results of record-wise bootstrapping with $k = k' = 50\,\%$ and $r = 10$ for the synthetic dataset $\boldsymbol{y}_{3,4}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 20 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

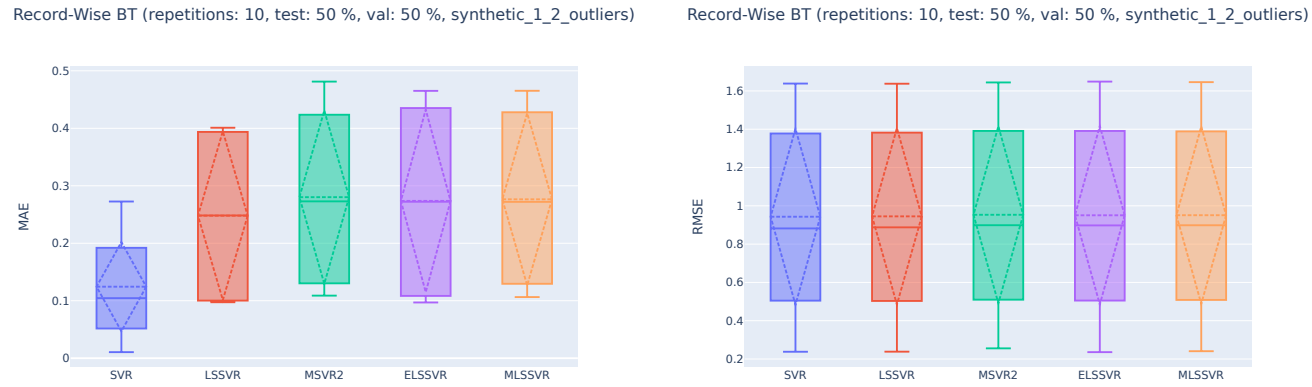


Figure 3: Results of record-wise bootstrapping with $k = k' = 50\,\%$ and $r = 10$ for the synthetic dataset $\boldsymbol{y}_{1,2}^{\text{outliers}}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 20 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

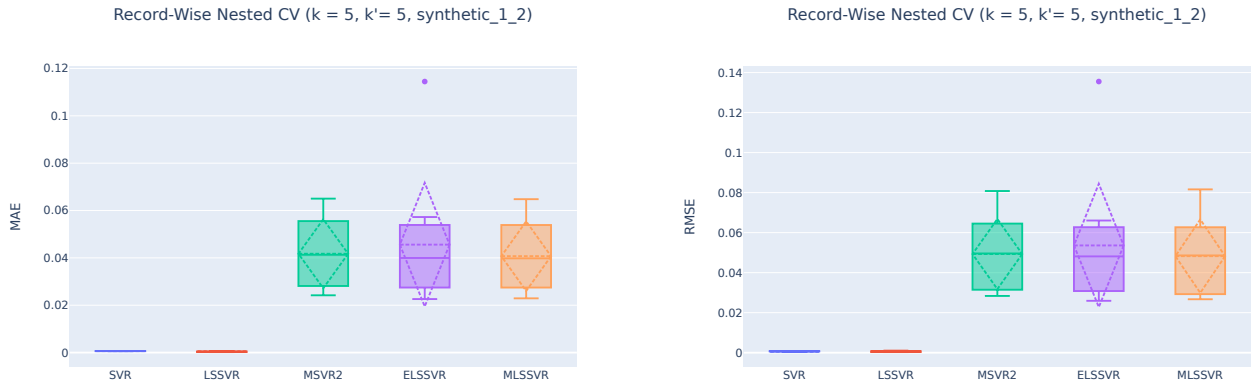


Figure 4: Results of record-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{1,2}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

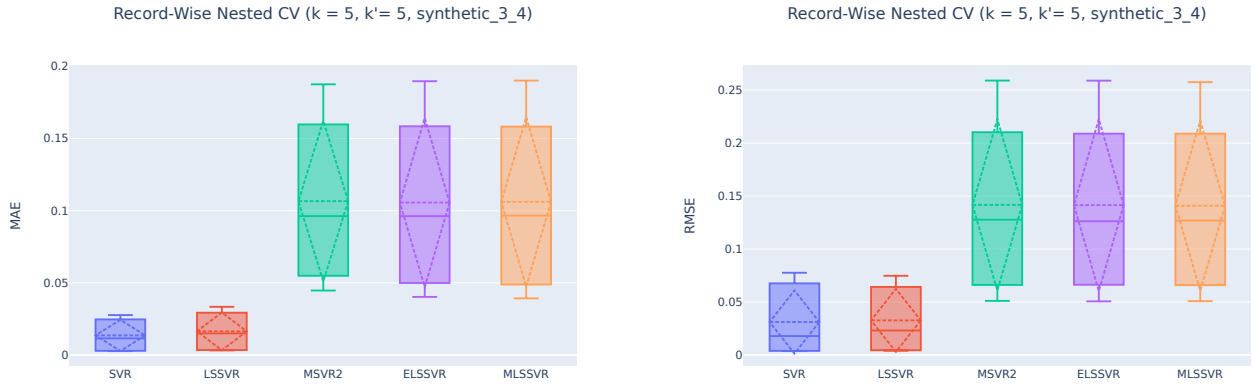


Figure 5: Results of record-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{3,4}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

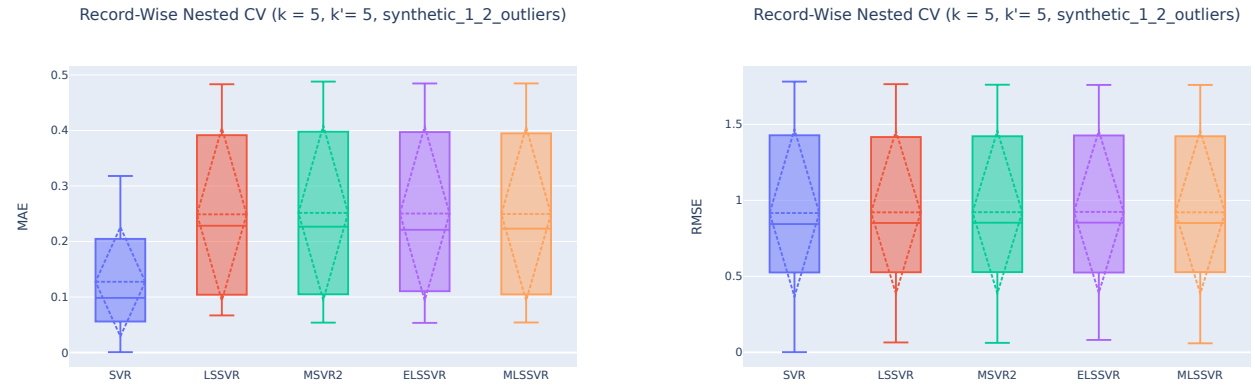


Figure 6: Results of record-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{1,2}^{\text{outliers}}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

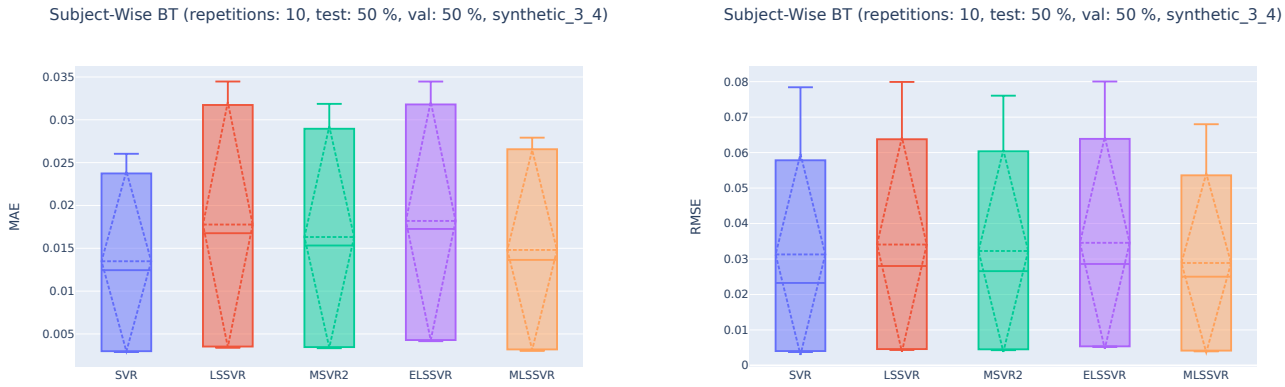


Figure 7: Results of subject-wise bootstrapping with $k = k' = 50\,\%$ and $r = 10$ for the synthetic dataset $\boldsymbol{y}_{3,4}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

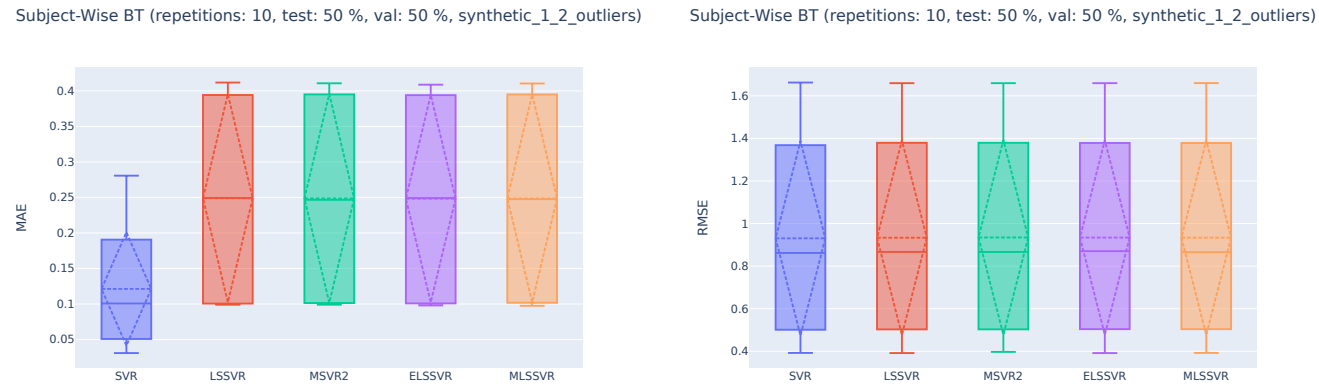


Figure 8: Results of subject-wise bootstrapping with $k = k' = 50\,\%$ and $r = 10$ for the synthetic dataset $\boldsymbol{y}_{1,2}^{\text{outliers}}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

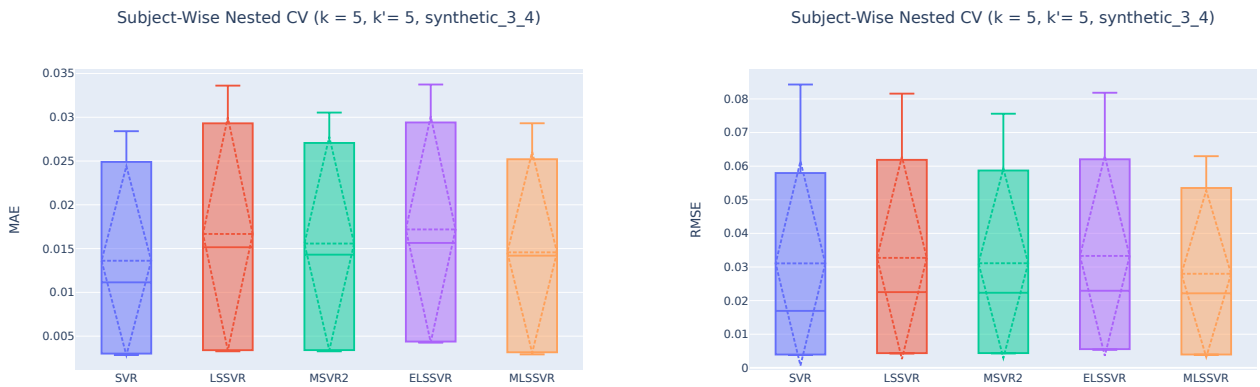


Figure 9: Results of subject-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{3,4}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

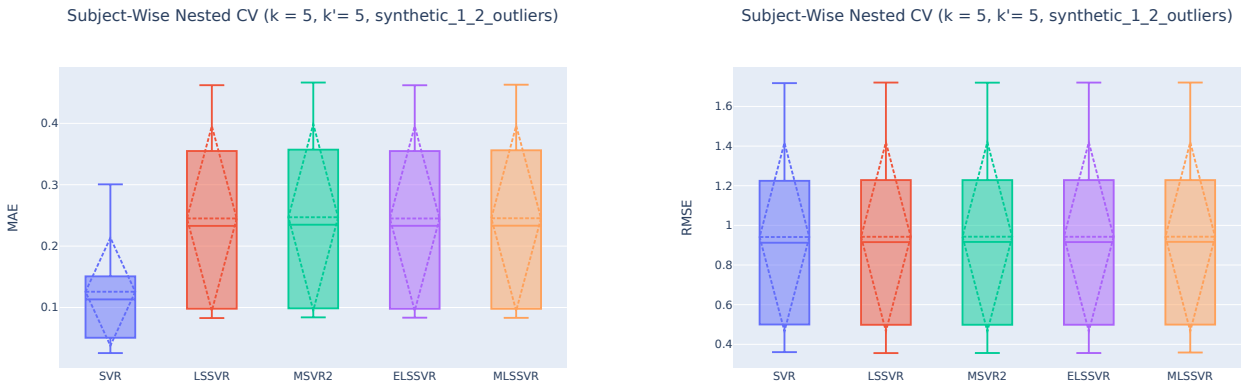


Figure 10: Results of subject-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{1,2}^{\text{outliers}}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

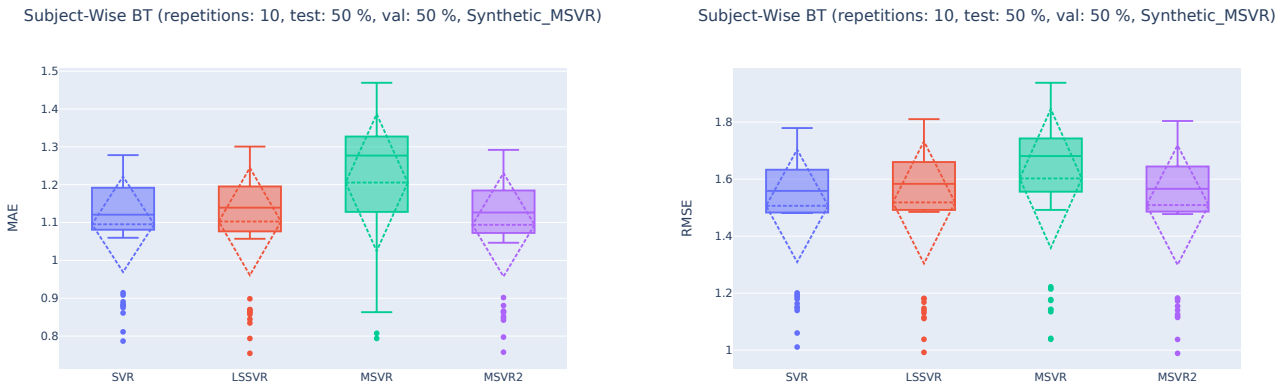


Figure 11: Results of subject-wise bootstrapping with $k = k' = 5$ and $r = 10$ for the synthetic M-SVR dataset. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of all five outputs, i.e., each boxplot is plotted for 50 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

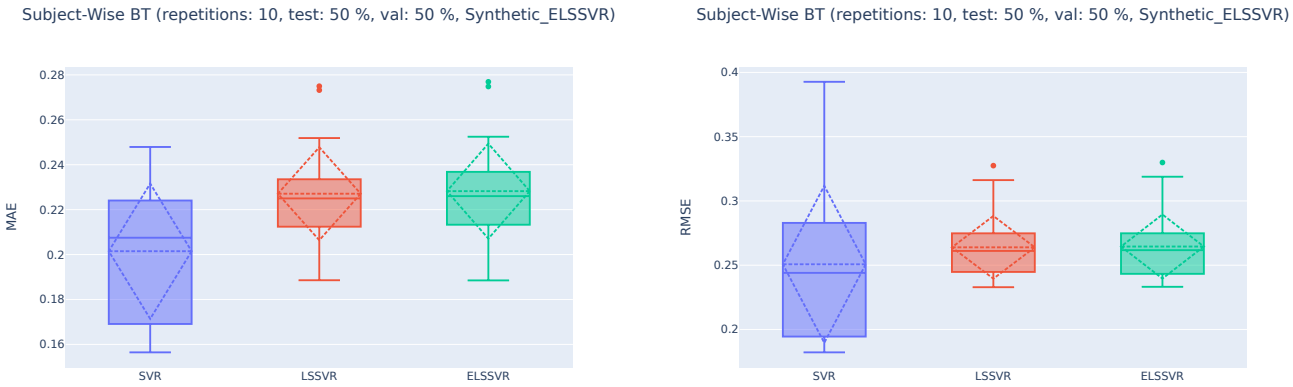


Figure 12: Results of subject-wise bootstrapping with $k = k' = 5$ and $r = 10$ for the synthetic ELS-SVR dataset. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 20 MAEs or RMSEs. The additional dashed lines are mean and standard deviation.

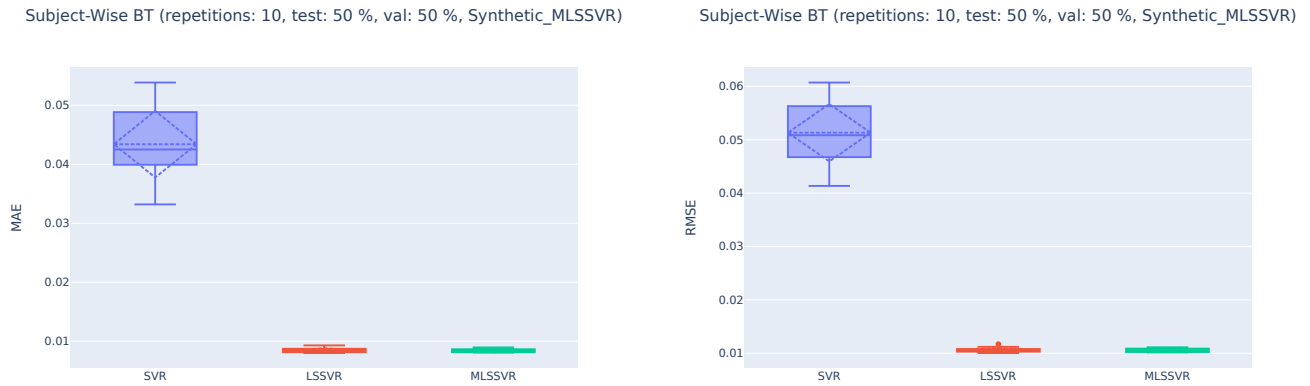


Figure 13: Results of subject-wise bootstrapping with $k = k' = 5$ and $r = 10$ for the synthetic MLS-SVR dataset. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 20 MAEs or RMSEs. The additional dashed lines are mean and standard deviation. Note that Scikit-learn's SVR could not consider $C > 2^7$ during hyperparameter tuning.

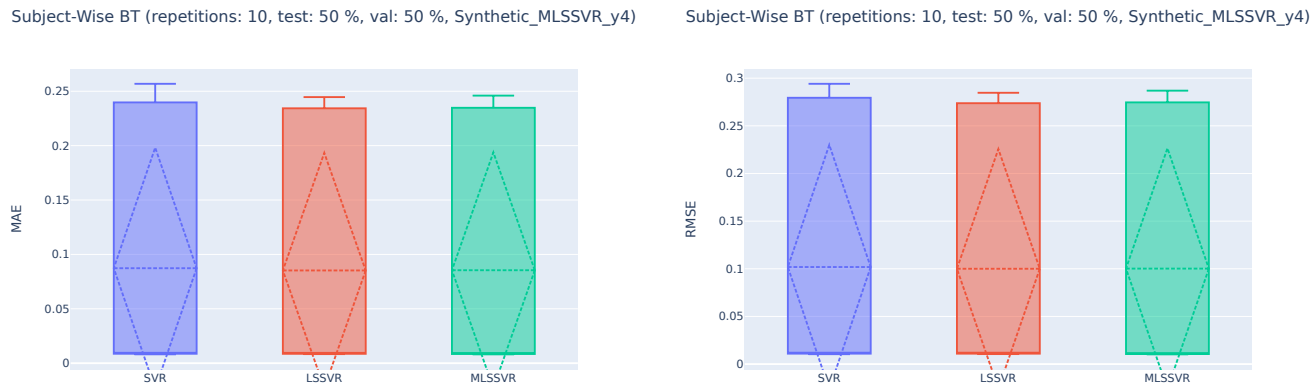


Figure 14: Results of subject-wise bootstrapping with $k = k' = 5$ and $r = 10$ for the synthetic MLS-SVR dataset with additional output $\boldsymbol{y_4}$. A boxplot for one method contains $r = 10$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of all three outputs, i.e., each boxplot is plotted for 30 MAEs or RMSEs. The additional dashed lines are mean and standard deviation. Note that Scikit-learn's SVR could not consider $C > 2^7$ during hyperparameter tuning.

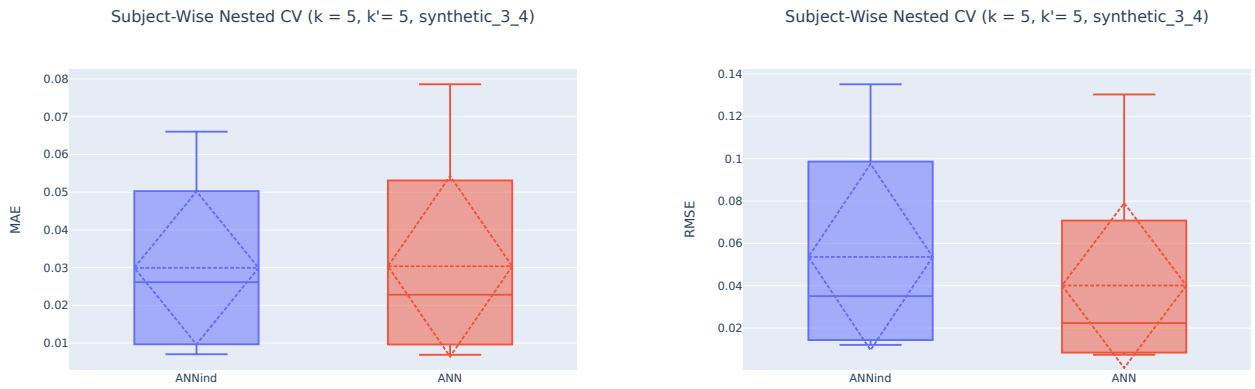


Figure 15: Results of subject-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{3,4}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation. "ANNind" stands for independent ANNs, i.e., the single-output approach, while "ANN" stands for the multi-output approach.

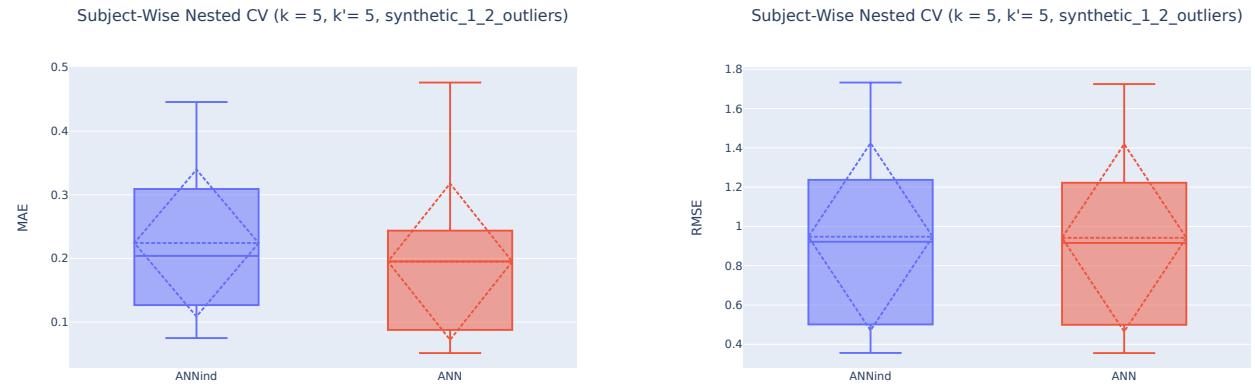


Figure 16: Results of subject-wise nested CV with $k = k' = 5$ for the synthetic dataset $\boldsymbol{y}_{1,2}^{\text{outliers}}$. A boxplot for one method contains $k = 5$ performances, where each performance consists of the MAEs (left) or RMSEs (right) of both outputs, i.e., each boxplot is plotted for 10 MAEs or RMSEs. The additional dashed lines are mean and standard deviation. "ANNind" stands for independent ANNs, i.e., the single-output approach, while "ANN" stands for the multi-output approach.

3.2 Drug response prediction: A critical systematic review of current datasets and methods

This section provides a brief background and outlines the motivation behind the second publication. The publication itself is included thereafter.

3.2.1 Background and motivation

As demonstrated in the first publication, multi-output SVR models were unable to leverage correlations among outputs to improve predictive performance, performing no better or even worse than single-output models. Unlike SVR, multi-output NNs did show improved performance over their single-output counterparts, prompting a shift in focus toward NNs.

A systematic review from 2023 [54] compared the performance of various NN-based models for drug response prediction. These models include autoencoders, attention mechanisms, deep NNs, visible NNs, convolutional NNs, and GNNs. The overall best-performing model at the time was the Twin Graph Neural Networks with Similarity Augmentation (TGSA) [55].

To develop a model that surpasses TGSA, a deep understanding of both TGSA itself and the datasets from DepMap is required. These datasets commonly provide MUT, CNV, and EXP data as input features, with IC_{50} or AUC values used as output metrics for drug response prediction. TGSA processes this input by constructing PPI networks for each cell line based on their MUT, CNV, and EXP features. A GNN is then trained on the PPI networks to produce cell line representations. In parallel, a second GNN is trained on molecular graphs to derive corresponding drug representations. Finally, TGSA combines the learned cell line and drug representations using a multilayer perceptron (MLP) to predict the IC_{50}, aiming to model how the interaction between a drug and the specific molecular profile of a cell line translates into a measurable response.

To determine a reference point for how well a new model would need to perform to surpass TGSA, the performance of TGSA was put into context by conducting basic baseline tests as well as comparisons against a simple multi-output multilayer perceptron (MMLP), and the DepMap datasets were thoroughly analyzed—once again yielding surprising results.

3.2.2 Publication

Pattern Recognition Letters 199 (2026) 21–26

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Drug response prediction: A critical systematic review of current datasets and methods

Nguyen Khoa Tran *, Gunnar W. Klau

Heinrich Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, North Rhine-Westphalia, Germany
Center for Digital Medicine, Düsseldorf, Germany

ARTICLE INFO

Editor: Doulaye Dembele

Dataset link: https://github.com/AlBi-HHU/Drug_Response_Prediction

Keywords:
Drug response prediction
Multi-output regression
Molecular graphs
Multi-omics
Multilayer perceptron
Graph neural network

ABSTRACT

Predicting drug response is a critical task in personalized medicine. Several recent studies have reported promising improvements in predictive performance with deep learning models trained on molecular characterizations of cell lines and drugs. However, our baseline tests suggest that little to no meaningful biological or chemical information is being learned from multi-omics data in the publicly available large-scale datasets GDSC and DepMap Public or molecular graphs, respectively. In our experiments, even gene expression data, commonly regarded as highly predictive, failed to deliver satisfactory drug response predictions. This raises the possibility that drug response measures or patterns observed in multi-omics data may not arise from underlying biological mechanisms. To investigate this, we identified and examined inconsistencies within and across the GDSC2 and DepMap Public 24Q2 datasets. We found that IC_{50} and AUC values of replicated experiments in GDSC2 had an average Pearson correlation coefficient of only 0.563 ± 0.230 and 0.468 ± 0.358, respectively. Additionally, somatic mutations shared between cell lines in the two datasets showed a Pearson correlation coefficient of only 0.180. Even in cases where TGSA, the current best-performing method to our knowledge, exceeded baseline performance, it still did not surpass a simple baseline multi-output multilayer perceptron (MMLP). Moreover, MMLP is not only more easily adaptable to new datasets but also significantly faster, making it a viable baseline for comparisons. In conclusion, our findings suggest that current cell-line and drug data are insufficient for existing modeling approaches to effectively uncover the biological and chemical mechanisms underlying drug response. Therefore, improving data quality or focusing on different data types is crucial before proposing novel methods.

1. Introduction

Precision medicine aims to tailor cancer therapies to individual patients, yet predicting a patient's response to a drug based on their biological characteristics remains challenging. This difficulty stems from the complex nature of cancer and the limited availability of clinical data.

To address these challenges, large-scale initiatives such as the Genomics of Drug Sensitivity in Cancer (GDSC) [1] and the Dependency Map (DepMap) project [2] have emerged. These projects involve testing numerous anti-cancer drugs across diverse cancer cell lines using high-throughput screening technologies. Additionally, these datasets include detailed gene profiles from omics data such as somatic mutations (MUT), copy number variations (CNV), and gene expression (EXP).

Drug response prediction models aim to find a mapping $f(x) \approx y$ from molecular characterizations of cell lines and drugs x to drug response values y. These response values are commonly measured as the half-maximal inhibitory concentration (IC_{50}) or the area under the curve (AUC). Machine learning models designed to predict drug response often include dense neural networks (DNNs) [3], convolutional neural networks (CNNs) [4–6], autoencoders [7], and attention mechanisms [8]. Other approaches involve random forests [9]. To further improve predictive accuracy, additional biological data like pathway information or protein-protein interaction (PPI) networks can be integrated through fully connected neural networks (FCNNs) [10, 11] or graph neural networks (GNNs) [12]. Drug features, including molecular fingerprints, drug targets, SMILES, and molecular graphs, have also been incorporated using DNNs [10], CNNs [4,5,8], and GNNs [6,12].

Eckhart et al. [13] have shown that among gene profiles, EXP generally offers better predictive power than MUT or CNV, though

* Corresponding author at: Heinrich Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, North Rhine-Westphalia, Germany.
E-mail addresses: nguyen.tran@hhu.de (N.K. Tran), gunnar.klau@hhu.de (G.W. Klau).

https://doi.org/10.1016/j.patrec.2025.10.016
Received 2 April 2025; Received in revised form 18 August 2025; Accepted 25 October 2025
Available online 30 October 2025

it is less robust to biological and technical variability [14]. Furthermore, dimensionality reduction can significantly improve the performance of both simple and complex models [13]. When incorporating additional biological data, pathway information has shown limited benefit [11], while PPI networks have been reported to improve predictive accuracy [12]. In terms of chemical representation, drug target data has been shown to be inferior to fingerprints [11], while fingerprints [10] and SMILES [8] have been shown to be inferior to molecular graphs [15,16]. Regarding model types, black-box models like multilayer perceptrons (MLPs) tend to outperform more interpretable models like random forests [11] or attention mechanisms [15].

In compliance with all these findings, twin graph neural networks with similarity augmentation (TGSA) [12] employs black-box GNNs for both molecular graphs and PPI networks and has emerged as the top-performing model in recent evaluations [15,16]. TGSA is reported to perform well in the *leave pairs out* (LPO) scenario, where drug responses for missing cell line-drug pairs are predicted when all cell lines and drugs are present in both training and test data. However, its performance is noted to decline in both blind test scenarios—a trend observed across multiple models [6,11,12,15]. These scenarios are the *leave cell lines out* (LCO) scenario, where cell lines are present in test data only, and the *leave drugs out* (LDO) scenario, where drugs are present in test data only. Both scenarios reflect real-world clinical settings, where predicting responses for previously unseen cell lines or drugs is essential.

To explore the reasons for this performance drop, we designed several baseline models for comparison with TGSA. Surprisingly, in the LDO scenario, TGSA did not exceed baseline performance, and in the LCO scenario, it did not surpass a simple multi-output multilayer perceptron (MMLP). Strikingly, even in the LPO scenario, TGSA failed to surpass MMLP. To investigate further, we conducted an ablation study on the gene profiles (MUT, CNV, and EXP) as well as the molecular graphs to assess their individual contributions to predictive performance. Additionally, we analyzed the GDSC2 dataset (version 8.5) and compared it with the DepMap Public dataset (version 24Q2). Our analysis reveals that within and across the datasets, MUT, CNV, and drug reponse values exhibit only low to moderate concordance. We conclude that with current data, existing models are unable to effectively uncover meaningful biological and chemical mechanisms underlying drug response, emphasizing the critical importance of improving data quality or generating and focusing on alternative data types. Existing models can then be adjusted to them and reevaluated.

In order to facilitate the verification of future datasets and models in LPO/LCO/LDO scenarios, we provide a user-friendly benchmark environment using Snakemake [17], which includes all baseline and ablation tests and MMLP. Since MMLP closely resembles a standard MLP, it can be easily adapted to various kinds of data and offers a low runtime, and thus it can serve as an additional baseline model that future models need to surpass.

2. Methodology

This section provides a description of the datasets, the two models TGSA and MMLP, and the experimental setup used in this study.

2.1. Datasets

The input data for drug response prediction models consists of at least two matrices: a feature matrix $x \in \mathbb{R}^{n \times d}$ and a target matrix $y \in \mathbb{R}^{n \times p}$, where n is the number of cancer cell lines, d is the number of gene features, and p is the number of cancer drugs. Regarding the GDSC2 dataset, we derived four feature matrices and one target matrix from Cell Model Passports [18]. We excluded the proteomics matrix due to 38.6% missing values, and we chose not to impute with 0 since missing entries correspond to high q-values, signaling unreliable protein abundance measurements. The three remaining feature matrices are the somatic mutation matrix $x_{\text{MUT}} \in \{0,1\}^{934 \times 23189}$, the copy number variation matrix $x_{\text{CNV}} \in \{-2,-1,0,1,2\}^{934 \times 20669}$, and the gene expression matrix $x_{\text{EXP}} \in \mathbb{R}_+^{934 \times 37005}$. The target matrix is the $\log_{10}(\text{IC}_{50})$ matrix $y \in \mathbb{R}^{934 \times 184}$, with 9.7% of the cell line-drug pairs missing. We filtered out all drugs without PubChem ID to enable running TGSA on the SMILES retrieved from PubChem [19] using PubChemPy (version 1.0.4) [20]. Also, since TGSA's PPI network requires feature selection for feasible runtime, we adopted the list compiled by Zhu et al. [12] containing cancer-related genes according to COSMIC [21], resulting in reduced input matrices: $x_{\text{MUT}}^{\text{COSMIC}} \in \{0,1\}^{934 \times 658}$, $x_{\text{CNV}}^{\text{COSMIC}} \in \{-2,-1,0,1,2\}^{934 \times 658}$, and $x_{\text{EXP}}^{\text{COSMIC}} \in \mathbb{R}_+^{934 \times 658}$. Note that although the input matrices are of different types (binary, discrete, and continuous), both TGSA and MMLP are able process them simultaneously, owing to the flexible trainability of neural network edge weights. Further details on the data and preprocessing are provided in the Supplementary Material (Sections 1, 2, and 3).

2.2. Twin graph neural networks with similarity augmentation (TGSA)

Zhu et al. [12] designed TGSA to integrate both fine-grained (gene-level and atom-level) and coarse-grained (sample-level) information through two main steps: twin graph neural networks for drug response prediction (TGDRP) and a similarity augmentation module.

The TGDRP step involves two components. The first is a PPI network built from the feature matrices $x_{\text{MUT}}^{\text{COSMIC}}$, $x_{\text{CNV}}^{\text{COSMIC}}$, and $x_{\text{EXP}}^{\text{COSMIC}}$, using the detailed protein links file (version 11.0) from STRING [22], containing 400 GB gene interaction data. Each node in the PPI network represents a gene with the node features being taken from the feature matrices, and edges are drawn between nodes if the detailed protein links file contains an interaction between two genes above a predefined threshold (default threshold is 0.95). The second component is a molecular graph of the form $G = (V, E)$ built from the SMILES of a drug using RDKit (version 2022.09.1) [23]. Predictions for cell line-drug pairs are generated by training one GNN on the PPI network and another on the molecular graph, combining their results using a fully connected neural network (FCNN).

The second optional step, the similarity augmentation module, requires one cell line graph and one drug graph that are built as follows: Each cell line or drug is represented as a node, where each cell line is connected to the five most similar cell lines according to the Pearson correlation coefficient (PCC) of their EXP values, whereas each drug is connected to the five most similar drugs according to the Jaccard similarity of their fingerprints. To compute the final drug response prediction for each cell line-drug pair, the model parameters of the earlier trained TGDRP model are used for initializing the cell line graph and the drug graph. After that, one GNN is trained on the cell line graph and another GNN on the drug graph, with their results subsequently combined in an FCNN.

2.3. Multi-output multilayer perceptron (MMLP)

Because training a separate MLP for every single drug is time-consuming [13] and disregards potential correlations among targets [24], our MMLP predicts a fixed number of targets simultaneously for given cell lines. However, it cannot be used in the LDO scenario since the same fixed number of targets is needed for training and testing. For its input, MMLP uses one feature matrix along with the target matrix. Multiple feature matrices with the same cell lines can be concatenated into a single matrix, e.g., x_{MUT}, x_{CNV}, and x_{EXP} can be concatenated into $x \in \mathbb{R}_+^{934 \times 80863}$.

MMLP follows the architecture of a standard MLP, with an input layer of d nodes (one per feature), a configurable number of hidden layers with h nodes (h is a hyperparameter), and an output layer of p nodes (one per drug). Two simple modifications are introduced: (1) a sigmoid layer between the input and first hidden layer, assigning a weight between 0 and 1 to each feature, interpretable as a feature

importance score [25]; (2) an imputation mask that sets gradients to zero during backpropagation for missing values. Modification (1) allows for more granularity, especially for binary features, and also supports feature selection if desired, while modification (2) ensures that imputed values (occurring due to missing values in the target matrix and/or due to the LPO scenario) do not affect parameter updates.

After performing hyperparameter tuning using grid search, we selected the best hyperparameters for the GDSC2 dataset, which were used in all experiments and are as follows: number of hidden layers = 1, hidden size $h = 2048$, batch size = 8, activation function = LeakyReLU, dropout ratio = 0.5, optimization algorithm = Adam, learning rate = 0.0001, weight decay = 0, loss function = mean squared error (MSE), maximum epochs = 300, with early stopping after 10 epochs of no improvement.

2.4. Test configuration

To ensure reliable results, we applied k-fold cross-validation (CV) with $k = 5$. For each split, the training set was used to optimize model parameters, selecting the best performance on the validation set before measuring performance on the test set. Hyperparameters for TGSA were taken from Zhu et al. [12], while hyperparameters for MMLP were preselected, see Section 2.3.

We employed three types of CV: record-wise (LPO), subject-wise (LCO), and target-wise (LDO). LPO CV tests the ability to predict missing cell line-drug pairs when all cell lines and drugs are present in the training data, LCO CV evaluates predictions for unseen cell lines, and LDO CV assesses predictions for unseen drugs. See Fig. 1 for an illustration of the three data splitting options.

While Shen et al. [15] and Menden et al. [26] also train and test models on a single drug at a time, we omitted this approach for TGSA because training on the drug features of a single drug would be redundant since there is no difference between the drug features among all datapoints.

Note that due to missing data (9.7%) and the fact that the number of cell lines/drugs is not always divisible by 5, the CV splits do not always contain exactly 20% of the data.

2.5. Baseline tests

We conducted the following baseline tests:

1. Mean predictor: For each unseen cell line-drug pair (i, j), the drug response prediction is computed as the mean of:
   - LPO: the average response of cell line i to all seen drugs and the average response of all seen cell lines to drug j.
   - LCO: the average response of all seen cell lines to drug j.
   - LDO: the average response of cell line i to all seen drugs.

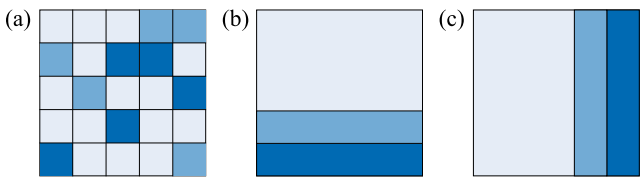


Fig. 1. An illustration of splitting target matrix y in three different ways: (a) record-wise (LPO), (b) subject-wise (LCO), or (c) target-wise (LDO). In this example, the training set (light blue) consists of 3/5 of y, while the validation set (medium blue) and test set (dark blue) consist of 1/5 each. Rows are cell lines and columns are drugs.

2. Biological information test: For models directly training on a feature matrix (e.g., MMLP), we replaced the feature matrix with an identity matrix of size $n \times n$ to eliminate any coincidental patterns or similarities between cell lines. This test is not applicable for LCO.
3. Chemical information test: For models training on molecular graphs (e.g., TGSA), we replaced the molecular graph of each drug i with a single node containing $p = 184$ node attributes. All attributes were set to 0 except for the ith attribute, ensuring the elimination of any coincidental patterns or similarities between drugs. This test is not applicable for LDO.
4. Shuffled feature matrix: We shuffled the entries of the feature matrix to check if any biological information is learned. As the shuffled data may introduce coincidental patterns that either favor or disadvantage the models, we preferred the second baseline test when testing for biological information and only ran this test for LCO.

3. Results

We implemented a benchmarking workflow for drug response prediction models using Python 3, PyTorch [27], and the Snakemake workflow management system [17]. The root mean square error (RMSE) was selected as the evaluation metric, as the commonly used coefficient of determination (R^2) is unsuitable for nonlinear models [28].

Figs. 2, 3, and 4 present the results for LPO, LCO, and LDO CV, respectively. For each model, the five RMSEs (one per test set) are displayed in a boxplot. Runtime for the TGSA models varied between 10 h and 3 days, while MMLP models took between 1 and 4 h. In the following, we distinguish between TGDRP (the first step of TGSA) and TGSA (both steps of TGSA). MMLP_{ALL} refers to MMLP trained on all features, i.e., the concatenation of x_{MUT}, x_{CNV}, and x_{EXP}. For models trained on individual feature matrices, MMLP_i represents MMLP trained on x_i, where $i \in \{\text{MUT}, \text{CNV}, \text{EXP}\}$. The superscript $^{\text{COSMIC}}$ is added when training on COSMIC genes only, e.g., $\text{MMLP}_{\text{ALL}}^{\text{COSMIC}}$. The second baseline test, the biological information test, is denoted as MMLP_{id}, while the third baseline tests, the chemical information tests, with TGDRP and TGSA are written as TGDRP_{id} and TGSA_{id}, respectively. The fourth baseline test, the shuffled feature matrix, was conducted on MMLP_{EXP} only and is referred to as $\text{MMLP}_{\text{EXP}}^{\text{shuffled}}$.

In the LPO scenario (Fig. 2), the best-performing model is $\text{MMLP}_{\text{ALL}}^{\text{COSMIC}}$, outperforming TGDRP and TGSA when trained on the same COSMIC genes. Selecting only COSMIC genes seems to generally have a positive effect on the performance of MMLP in the LPO scenario. Even though $\text{MMLP}_{\text{ALL}}^{\text{COSMIC}}$ and other models demonstrate low RMSEs, none significantly outperform the baseline tests. TGDRP and TGSA achieve mean RMSEs of 0.936 and 0.937, respectively, slightly worse than the baseline TGSA_{id} (mean RMSE: 0.935), suggesting no meaningful chemical information was learned. $\text{MMLP}_{\text{ALL}}^{\text{COSMIC}}$ with a mean RMSE of 0.930 shows a marginal improvement of 4% ($= 1 - \frac{\text{mean RMSE of model}}{\text{mean RMSE of baseline test}}$) over MMLP_{id} (mean RMSE: 0.970), indicating limited biological information is learned. TGDRP and TGSA perform worse than $\text{MMLP}_{\text{ALL}}^{\text{COSMIC}}$, suggesting they also fail to capture significant biological insights from the PPI network. SAURON-RF [9], which is, to the best of our knowledge, the best-performing random forest model, was evaluated by the original authors in the LPO scenario with 5-fold CV, but only on a large subset of all cell line-drug pairs of the GDSC dataset and only on 20 to 100 EXP features due to scalability limitations. The authors reported an MSE of 1.96 (corresponding to an RMSE of 1.4), which is significantly worse than all other methods except the mean predictor. This is consistent with the findings in [11], where black-box models such as MLPs generally outperformed more interpretable models like random forests. Hence, we excluded random forests from further analyses.

In the LCO scenario (Fig. 3), MMLP_{EXP} achieves the best performance with a mean RMSE of 1.283. Once again, TGDRP and TGSA perform worse than the baseline TGSA_{id} (mean RMSE: 1.341), reinforcing

the hypothesis that no chemical information is learned from molecular graphs. However, as many models outperform the first and fourth baselines (mean predictor and shuffled x_{EXP}), with MMLP_{EXP} outperforming by 16%, some biological information seems to be captured, especially from EXP data. In contrast to the LPO scenario, COSMIC gene selection in LCO negatively impacts MMLP_{ALL} and MMLP_{EXP}.

In the LDO scenario (Fig. 4), TGDRP emerges as the best-performing model. However, Shen et al. [15] reported a significantly higher mean RMSE for TGSA (approximately 2.7, with the exact value provided upon request being 2.6642) compared to our obtained mean RMSE. Therefore, we repeated 5-fold CV five times. Overall, the resulting RMSEs of LPO, LCO, and LDO for TGDRP and TGSA indicate that TGSA performs similarly to TGDRP, but not better. The results from LPO and LCO repetitions were consistent with Figs. 2 and 3, so they

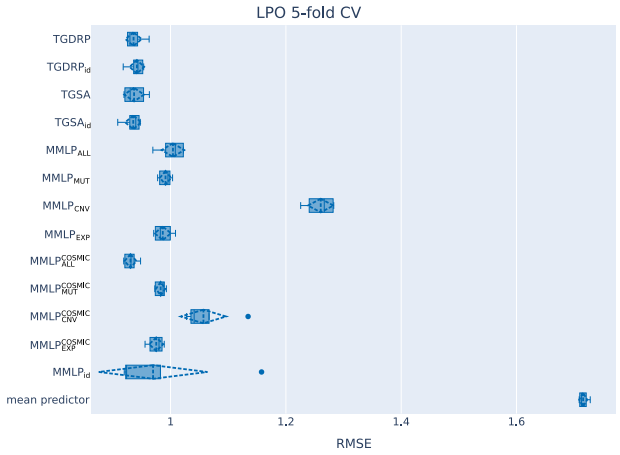


Fig. 2. LPO 5-fold CV results for GDSC2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation.

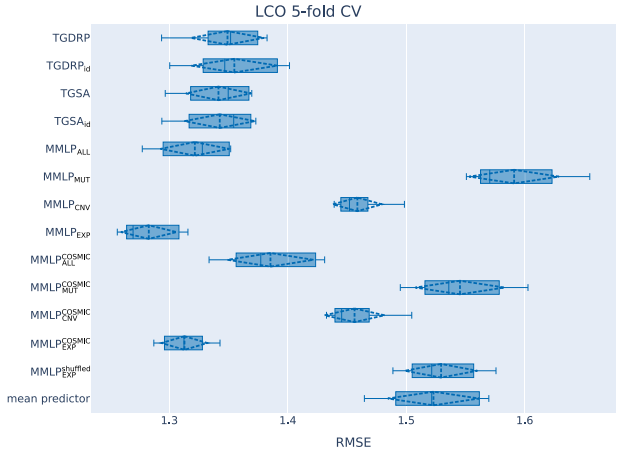


Fig. 3. LCO 5-fold CV results for GDSC2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation. Note that for the fourth baseline test, only the result for $\text{MMLP}_{\text{EXP}}^{\text{shuffled}}$ is shown because MMLP_{EXP} performed best.

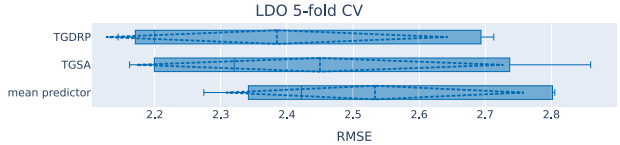


Fig. 4. LDO 5-fold CV results for GDSC2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation.

are omitted. However, the LDO results showed considerable variation across data splits. The average RMSE and standard deviation across five repetitions for TGDRP, TGSA, and the mean predictor are 2.487 ± 0.280, 2.538 ± 0.280, and 2.536 ± 0.242, respectively. Strikingly, the mean predictor outperformed TGSA and was only 2% worse than TGDRP while having a 14% more stable standard deviation. Given that the RMSEs for TGDRP and TGSA are approximately 2.5, and considering that the drug response values are $\log_{10}$-transformed, this translates to predicted values that are roughly 300 ($\approx 10^{2.5}$) times the original drug response or about $\frac{1}{300}$ of it. Hence, we conclude that the LDO scenario remains inadequately addressed by current methods.

For the smaller DepMap Public 24Q2 dataset with 474 cell lines and 24 drugs, TGDRP is clearly outperformed by MMLP (LPO and LCO) and the mean predictor (LDO), see the Supplementary Material (Section 4).

4. Discussion

We selected GDSC2 for our experiments due to its improved screening methodologies compared to GDSC1 [1], and its larger set of cell lines and drugs than DepMap Public 24Q2. Given the poor predictive performance, we explored whether data quality might also play a role. We used the concordance correlation coefficient (CCC) instead of the widely used Pearson correlation coefficient (PCC) to evaluate data reproducibility, because CCC accounts for both correlation and agreement, unlike PCC, which measures only linear correlation. The CCC is defined as $\text{CCC} = \frac{2\text{PCC}\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2}$, where x and y are the two variables with means μ_x and μ_y as well as variances σ_x^2 and σ_y^2, respectively. While the PCC is usually reported with a p-value representing the probability that, if the true correlation were zero, a dataset would yield a PCC at least as far from zero as the observed value in either direction (i.e., a two-sided test) [29], the CCC is usually reported with a 95% confidence interval (CI), often obtained via bootstrap; in our analysis, we used 100 repetitions. The CI reflects the range of plausible values for the true concordance between two datasets and is used instead of a p-value because, unlike a PCC of zero resulting only from no linear correlation, a CCC of (practically) zero can result from substantial bias (e.g., $x = y + 9999$), scale differences (e.g., $x = 1000 \cdot y$), perfect negative correlation, or a combination of these factors, making it difficult to define a single, meaningful null hypothesis.

First, we examined the drug response values, i.e., IC_{50} or AUC values, by comparing duplicate cell line-drug experiments in GDSC2. 9 drugs were tested twice on up to 792 cell lines, resulting in 6288 duplicate cell line-drug pairs. For each drug, we calculated the CCC among IC_{50} values and among AUC values of duplicate experiments, with an average CCC of $0.563\% \pm 0.230$ for IC_{50} and $0.468\% \pm 0.358$ for AUC. While four drugs showed a CCC above 0.7 for both IC_{50} and AUC, three drugs had CCCs below 0.3, even down to 0.013 (CI: [0.010, 0.016]), suggesting that the drug screenings are inconsistent when reproduced. This is further supported by comparisons between IC_{50} values from 368 cell lines and 9 drugs GDSC2 and DepMap Public 24Q2 have in common, which yield a CCC of only 0.409 (CI: [0.388, 0.430]), although this outcome might be influenced as GDSC2 and DepMap Public 24Q2 use different concentration ranges in the drug screenings. Coupled with criticisms of IC_{50} and AUC as drug response metrics such as incomparability across drugs, dependence on concentration range, and ignoring proliferation rate of cell lines [30], these findings highlight the need for different measures, which are currently not included in the GDSC2 and DepMap Public 24Q2 datasets. Potential alternatives include growth rate inhibition (GR) [30] or metrics derived from live-cell imaging data. Like traditional drug screening, live-cell imaging incubates cell lines at a few specific drug concentrations, but instead of providing cell viability at a single time point, it captures images over a range of time points, allowing cell viability to be indirectly inferred by counting cells and offering greater robustness due to the availability of more data points.

Next, we investigated the omics data. Comparing the MUT file and the pan-cancer gene feature file, both from GDSC2 (920 cell lines and 285 genes in common), revealed low overlap, with only 6.0% of mutations shared, and a CCC of 0.084 (CI: [0.082, 0.087]). A similar low overlap is between MUT data of GDSC2 and DepMap Public 24Q2 (1213 cell lines and 18 300 genes in common), with an overlap of 11.7% and a CCC of 0.180 (CI: [0.179, 0.180]). This is consistent with the findings by Ben-David et al. [31] that, although cell lines are generally considered clonal, they are in fact highly genetically heterogeneous, leading to variable drug responses. As DepMap Public 24Q2 measures the relative copy number whereas the CNV data of GDSC2 is categorical, CNV data between the two datasets (941 cell lines and 19 097 genes in common), could not be directly compared with the CCC, so instead we calculated a PCC of 0.519 (p-value $<5 \cdot 10^{-324}$), indicating moderate correlation. Surprisingly, gene expression data measured in transcripts per million showed high agreement (CCC: 0.951, CI: [0.951, 0.952]) between GDSC2 and DepMap Public 24Q2 (998 cell lines and 19 061 genes in common), despite being considered a less reliable biomarker than mutation data due to biological and technical factors [14]. This finding should be explored in future studies. Given these results, we opted not to merge GDSC2 and DepMap Public 24Q2 datasets unlike Zhu et al. [12], as their feature and target matrices lacked strong correlations overall. Altogether, these observations align with our LCO results (Fig. 3), where MMLP_{EXP} and $\text{MMLP}_{\text{EXP}}^{\text{COSMIC}}$ performed best while MMLP_{MUT} and $\text{MMLP}_{\text{MUT}}^{\text{COSMIC}}$ performed worst overall, implying that biological information can be learned from EXP but not MUT. Nevertheless, in the LPO results (Fig. 2), neither MMLP_{EXP} nor $\text{MMLP}_{\text{EXP}}^{\text{COSMIC}}$ surpassed the baseline MMLP_{id}, which uses no biological information at all. This suggests that while EXP provides some valuable information, it is insufficient on its own to give accurate drug response predictions, as even in the LCO results, MMLP_{EXP} predicts values roughly 19 ($\approx 10^{1.28}$) times the original drug response or about $\frac{1}{19}$ of it. Instead, predictions seem mostly influenced by patterns across the drug responses for a cell line, which would also explain the poor results in the LDO scenario (Fig. 4).

Most drugs exert their effects by binding to target proteins [32]. Transcriptomic EXP data serve as a proxy for gene function by capturing how actively a gene is transcribed into messenger RNA (mRNA), an intermediate in protein production. As such, EXP data may outperform genomic MUT and CNV data for drug response prediction. Proteomics data may be even more effective, as they provide direct insights into produced proteins and cellular processes [33]. However, the current usability of proteomics is hindered by missing values, as noted in Section 2.1, highlighting the urgent need to develop methods that ensure comprehensive measurement of proteomic data.

To capture the full complexity of cell line-drug interactions, both suitable biological and chemical features must be effectively integrated. Biological features should be modeled using methods suited to their inherent structure. For example, MUT, CNV, and EXP are unordered, making CNN-based approaches such as tCNNS [5] and GraphDRP [6] inappropriate. Additionally, these data are non-sequential, making recently highly successful sequential models such as recurrent neural networks (RNNs) or transformers inappropriate as well. Regarding chemical data, all current methods struggle to effectively incorporate drug features into drug response prediction models. Although the use of GNNs on molecular graphs to derive drug embeddings appears conceptually sound at first, the poor performance of TGSA suggests that intramolecular signaling among atoms either lacks meaningful chemical basis or fails to contribute useful information. One possible explanation is that most drugs exert their effects by binding to specific target proteins expressed by the cell, a process that is highly dependent on the three-dimensional (3D) structures of both the drug and the protein. Molecular graphs, however, represent only two-dimensional (2D) connectivity and lack explicit 3D structural information, preventing GNNs from capturing drug-target binding interactions, which would also explain why PPI networks did not improve predictive performance. This fundamental limitation also extends to other currently used drug representations such as fingerprints or SMILES, which lack 3D spatial data as well. We therefore propose that future work should focus on predicting drug-target binding directly by leveraging experimentally resolved or predicted protein structures (e.g., via AlphaFold [34,35]) alongside known or computed chemical structures (e.g., via RDKit [23]). These structural representations can serve as inputs to drug-target prediction tools such as DiffDock [36], enabling the prediction of whether and how a drug is likely to bind to a given target, and subsequently, whether such binding leads to a specific cellular response such as growth inhibition, growth stimulation, or cell death. An alternative direction is to revisit the integration of drug-target data. Although it has been reported to perform worse than molecular fingerprints [11], this may be attributed to the large number of unknown or poorly characterized drug-target interactions, as well as the discontinuation of support for the STITCH database [37] since 2015.

Lastly, we observed in the results section that selecting only COSMIC genes improved performance in the LPO scenario but achieved the opposite in the LCO scenario. The improvement in LPO suggests that excluding non-COSMIC genes may prevent overfitting or reduce noise by focusing on important genes only. However, since training on all genes resulted in higher RMSEs for the validation sets, overfitting can be ruled out. The decline in LCO suggests that limiting the model to known cancer-related genes may exclude important predictive features. These two seemingly contradictory findings suggest that a more refined feature selection strategy could enhance performance in both scenarios by balancing feature inclusion and exclusion. However, the excessive runtime of TGSA prevented a comparison with MMLP in feature selection experiments. Furthermore, given the previously discussed inherent data incongruence, such a comparison would likely provide limited insight.

5. Conclusions

We benchmarked the state-of-the-art TGSA model against several baseline tests and MMLP, a simple multi-output multilayer perceptron, across different CV scenarios. While both TGSA and MMLP struggled to learn meaningful biological and/or chemical information, MMLP consistently outperformed TGSA in terms of RMSE in the LPO and LCO scenarios with much shorter runtimes. However, the LDO scenario remains inadequately addressed by current models.

Our findings emphasize the critical need to refine both the acquisition and selection of gene input data, employing more reliable drug response metrics, and improving methods for incorporating chemical information before proposing complex, innovative models. Current and new models can then be adapted or developed for these data sources and should be systematically (re)assessed. For future benchmarking efforts, we recommend conducting baseline tests to ensure no spurious biological or chemical information influences results and to use MMLP as an additional baseline model.

CRediT authorship contribution statement

Nguyen Khoa Tran: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gunnar W. Klau:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Max J. Ried for the provided computational infrastructure and support as well as Torben Schmitz and Marvin Gooßens for the implementation of the imputation mask during backpropagation and the sigmoid layer after the input layer.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2025.10.016.

Data availability

Code, data, and links to data are found on https://github.com/AlBi-HHU/Drug_Response_Prediction.

References

[1] W. Yang, J. Soares, P. Greninger, et al., Genomics of drug sensitivity in cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells, Nucleic Acids Res. 41 (D1) (2012) D955–D961.
[2] R. Arafeh, T. Shibue, J. Dempster, et al., The present and future of the cancer dependency map, Nat. Rev. Cancer 25 (1) (2025) 59–73, URL: https://doi.org/10.1038/s41568-024-00763-x.
[3] Y. Chiu, H. Chen, T. Zhang, et al., Predicting drug response of tumors from integrated genomic profiles by deep neural networks, BMC Med. Genom. 12 (2019) 143–155.
[4] Y. Chang, H. Park, H. Yang, et al., Cancer drug response profile scan (CDRscan): a deep learning model that predicts drug effectiveness from cancer genomic signature, Sci. Rep. 8 (1) (2018) 8857.
[5] P. Liu, H. Li, S. Li, et al., Improving prediction of phenotypic drug response on cancer cell lines using deep convolutional network, BMC Bioinformatics 20 (2019) 1–14.
[6] T. Nguyen, G. Nguyen, T. Nguyen, et al., Graph convolutional networks for drug response prediction, IEEE/ACM Trans. Comput. Biol. Bioinform. 19 (1) (2021) 146–154.
[7] M. Li, Y. Wang, R. Zheng, et al., DeepDSC: a deep learning method to predict drug sensitivity of cancer cell lines, IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (2) (2019) 575–582.
[8] M. Manica, A. Oskooei, J. Born, et al., Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders, Mol. Pharm. 16 (12) (2019) 4797–4806.
[9] K. Lenhof, L. Eckhart, N. Gerstner, et al., Simultaneous regression and classification for drug sensitivity prediction using an advanced random forest method, Sci. Rep. 12 (1) (2022) 13458.
[10] B. Kuenzi, J. Park, S. Fong, et al., Predicting drug response and synergy using a deep learning model of human cancer cells, Cancer Cell 38 (5) (2020) 672–684.
[11] Y. Li, D. Hostallero, A. Emad, Interpretable deep learning architectures for improving drug response prediction performance: myth or reality? Bioinform. 39 (6) (2023) btad390.
[12] Y. Zhu, Z. Ouyang, W. Chen, et al., TGSA: protein–protein association-based twin graph neural networks for drug response prediction with similarity augmentation, Bioinform. 38 (2) (2022) 461–468.
[13] L. Eckhart, K. Lenhof, L. Rolli, et al., A comprehensive benchmarking of machine learning algorithms and dimensionality reduction methods for drug sensitivity prediction, Brief. Bioinform. 25 (4) (2024) bbae242.
[14] S. Mourragui, M. Loog, M. van Nee, et al., Percolate: an exponential family jive model to design dna-based predictors of drug response, in: International Conference on Research in Computational Molecular Biology, Springer, 2023, pp. 120–138.
[15] B. Shen, F. Feng, K. Li, et al., A systematic assessment of deep learning methods for drug response prediction: from in vitro to clinical applications, Brief. Bioinform. 24 (1) (2023) bbac605.
[16] C. Lin, Y. Guan, H. Li, Artificial intelligence approaches for molecular representation in drug response prediction, Curr. Opin. Struct. Biol. 84 (2024) 102747.
[17] F. Mölder, K. Jablonski, B. Letcher, et al., Sustainable data analysis with snakemake, F1000Research 10 (2021).
[18] D. van der Meer, S. Barthorpe, W. Yang, et al., Cell model passports—a hub for clinical, genetic and functional datasets of preclinical cancer models, Nucleic Acids Res. 47 (D1) (2019) D923–D929.
[19] S. Kim, J. Chen, T. Cheng, et al., PubChem 2023 update, Nucleic Acids Res. 51 (D1) (2023) D1373–D1380.
[20] P. Walters, PubChemPy: A python wrapper for the PubChem PUG REST API, 2014, https://github.com/mcs07/PubChemPy.
[21] J. Tate, S. Bamford, H. Jubb, et al., COSMIC: the catalogue of somatic mutations in cancer, Nucleic Acids Res. 47 (D1) (2019) D941–D947.
[22] D. Szklarczyk, R. Kirsch, M. Koutrouli, et al., The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest, Nucleic Acids Res. 51 (D1) (2023) D638–D646.
[23] G. Landrum, P. Tosco, B. Kelley, et al., Rdkit/rdkit: 2022_09_1 (Q3 2022) release (release_2022_09_1), 2022, http://dx.doi.org/10.5281/zenodo.7235579.
[24] N. Tran, L. Kühle, G. Klau, A critical review of multi-output support vector regression, Pattern Recognit. Lett. 178 (2024) 69–75.
[25] V. Borisov, J. Haug, G. Kasneci, CancelOut: A layer for feature selection in deep neural networks, in: Artificial Neural Networks and Machine Learning–ICANN 2019: Deep Learning: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part II 28, Springer, 2019, pp. 72–83.
[26] M. Menden, F. Iorio, M. Garnett, et al., Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties, PLoS One 8 (4) (2013) e61318.
[27] A. Paszke, S. Gross, F. Massa, et al., PyTorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).
[28] A. Spiess, N. Neumeyer, An evaluation of R^2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach, BMC Pharmacol. 10 (2010) 1–11.
[29] P. Virtanen, R. Gommers, T. Oliphant, et al., SciPy 1.0: Fundamental algorithms for scientific computing in python, Nature Methods 17 (2020) 261–272, URL: https://doi.org/10.1038/s41592-019-0686-2.
[30] M. Hafner, M. Niepel, M. Chung, et al., Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs, Nature Methods 13 (6) (2016) 521–527.
[31] U. Ben-David, B. Siranosian, G. Ha, et al., Genetic and transcriptional evolution alters cancer cell line drug response, Nat. 560 (7718) (2018) 325–330.
[32] B. Bryant, K. Knights, Pharmacology for Health Professionals, Elsevier Australia, 2011.
[33] N. Branson, P. Cutillas, C. Bessant, Comparison of multiple modalities for drug response prediction with learning curves using neural networks and XGBoost, Bioinform. Adv. 4 (1) (2023) vbad190, http://dx.doi.org/10.1093/bioadv/vbad190.
[34] J. Jumper, R. Evans, A. Pritzel, et al., Highly accurate protein structure prediction with AlphaFold, Nat. 596 (7873) (2021) 583–589.
[35] M. Varadi, S. Anyango, M. Deshpande, et al., AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models, Nucleic Acids Res. 50 (D1) (2022) D439–D444.
[36] G. Corso, H. Stärk, B. Jing, R. Barzilay, T. Jaakkola, Diffdock: Diffusion steps, twists, and turns for molecular docking, 2022, arXiv preprint arXiv:2210.01776.
[37] D. Szklarczyk, A. Santos, C. Von Mering, et al., STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data, Nucleic Acids Res. 44 (D1) (2016) D380–D384.

Supplementary Material for: Drug response prediction: A critical systematic review of current datasets and methods

Nguyen Khoa Tran[a,b,*], Gunnar W. Klau[a,b]

[a]Heinrich Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, North Rhine-Westphalia, Germany
[b]Center for Digital Medicine, Düsseldorf, Germany

1. Preprocessing DepMap Public 24Q2 files

DepMap Public 24Q2 target file (from the Cancer Cell Line Encyclopedia (CCLE) subproject): The original file is in long format. We transformed it into wide format with cell line names as indices, PubChem IDs as columns, and IC_{50} values as values. Originally, the columns are denoted by the drug names, which we replaced with the first PubChem ID found by PubChemPy. This results in a 504×25 matrix.

DepMap Public 24Q2 MUT feature file: All rows stand for a mutation. Some cell lines have multiple mutations in a feature, which we ignore as we are only interested in whether the gene is mutated or not. The original file is in long format. We transformed it into wide format with cell line names as indices, gene names as columns, and the mutation status as values. NaN is treated as not mutated, any other entry (SNV, insertion, etc.) is treated as mutated. This results in a 1788×19450 matrix.

DepMap Public 24Q2 CNV feature file: The original file is in wide format already. We only needed to remove the additional gene information in each column after the space as well as remove the columns with at least one missing value. We chose to remove 31 columns instead of 416 rows containing at least one missing value to keep as much data as possible. This results in a 1788×24352 matrix.

DepMap Public 24Q2 EXP feature file: The original file is in wide format already. We only needed to remove the additional gene information in each column after the space. This results in a 1517×19193 matrix.

2. Preprocessing GDSC2 files

GDSC2 target files: The original file is in long format. To transform it into wide format, we removed duplicate experiments as follows: If a cell line-drug pair appears multiple times, we take the last pair (which is an arbitrary decision). Cell line names are indices, drug names are columns, and either IC50 or AUC are values, depending on which drug response metric is desired. This results in a 969×288 matrix for both IC_{50} and AUC. For TGSA, only drugs with a PubChem ID can be used, hence we filtered out all drugs that have "NaN", "none", or "several" as an entry in the PubChem column of GDSC2's Drug_list.csv. Rows with all entries being NaN are removed. This results in a 963×184 matrix for both IC_{50} and AUC.

GDSC2 pan-cancer feature file: The original file is in long format. We transformed it into wide format with cell line names as indices, gene features as columns, and the mutation status as values. We dropped rows that contain at least one NaN entry. This results in a 925×735 matrix.

GDSC2 MUT feature file: All rows stand for a mutation. Some cell lines have multiple mutations in a feature, which we ignore as we are only interested in whether the gene is mutated or not. The original file is in long format. We transformed it into wide format with cell line names as indices, gene names as columns, and the mutation status as values. This results in a 1435×23189 matrix.

GDSC2 CNV feature file: We decided on the GISTIC instead of the PICNIC file because PICNIC's values can have different meanings (e.g., 2 can be neutral or a loss), whereas GISTIC's values are easy to interpret (-2 means deletion, -1 loss, 0 neutral, 1 gain, 2 amplification). The original file is in wide format already. We only needed to remove the first few rows and columns to have cell line names as indices and gene names as columns. We dropped columns containing at least one NaN because dropping rows would have removed all rows. This results in a 978×20669 matrix.

GDSC2 EXP feature file: The original file is in wide format already. We only needed to remove the first few rows and columns to have cell line names as indices and gene names as columns. We dropped columns containing at least one NaN because dropping rows would have removed all rows. Furthermore, the columns EEF1AKNMT and SEPTIN4 are duplicated, we removed both copies of each. This results in a 1431×37005 matrix. We decided on the transcripts per million (TPM) unit to enable a comparison to DepMap Public 24Q2's gene expression file. Since DepMap Public 24Q2's gene expression file contains $\log_2(\text{TPM} + 1)$ values, we also $\log_2$-transformed the GDSC2 EXP values after adding 1. Note that while the Cell Model Passport documentation for GDSC states that expression data are reported as $\log_2(\text{TPM}+1)$, the downloaded values appeared to be untransformed TPMs. Only after manually applying a $\log_2(\text{TPM} + 1)$-transformation did the values fall within a similar range as the DepMap Public 24Q2 data.

In addition to the MUT, CNV, and EXP files, we created shuffled versions of them for the fourth baseline test by flattening each matrix, shuffling all entries, and then reshaping them into the original shape.

The final preprocessing step necessary for GDSC2 is to save the cell lines shared between all possible data subsets. This means that if for example we want to train on MUT only, we need to use only the cell lines shared by the MUT

*Corresponding author
nguyen.tran@hhu.de (N.K. Tran); gunnar.klau@hhu.de (G.W. Klau)
ORCID(s): 0000-0002-4732-4294 (N.K. Tran); 0000-0002-6340-0090 (G.W. Klau)

matrix and the target matrix, and if we want to train on MUT, CNV, and EXP, we need to use only the cell lines shared between all four matrices (MUT, CNV, EXP, target matrix). The number of shared cell lines slightly varies for all data subsets (e.g., the number of shared cell lines between target matrix and EXP only is 956, while the number of shared cell lines between target matrix and MUT, CNV, EXP together is 934, which is the number we reported in Section 2.1), however, for simplification purposes we did not mention this in the main body.

Moreover, we did not mention the following renaming of GDSC2 cell line names in the discussion, which was necessary to enable comparing as much data as possible between GDSC2 and DepMap Public 24Q2:

- KM-H2 was renamed to KMH2,
- KMH-2 was renamed to KMHDASH2,
- MS-1 was renamed to MSDASH1,
- T-T was renamed to TDOTT because it is named T.T in DepMap Public 24Q2.

Furthermore, we removed all special characters (“.”, “-”, “(”, “)”, “/”, “:”, “_”, “[”, “]”) from the cell line names and ignored upper and lower case.

3. Preprocessing TGSA files

For TGSA’s GNNs, we created a SMILES file according to the original authors’s procedure, containing one row for each drug with the columns being the drug name, the compound identifier, the canonical SMILES, and the isomeric SMILES.

Note that Zhu et al. [12] obtained 706 instead of 658 COSMIC genes, as they used DepMap Public 24Q2 feature files while we use GDSC2’s feature files, which do not contain all of these COSMIC genes.

4. Results on DepMap Public 24Q2 dataset

Analogously to GDSC2, we only use the 474 cell lines shared between the target, MUT, CNV, and EXP matrices of DepMap Public 24Q2. Furthermore, we preprocessed the TGSA files in the same way, but only 683 out of 706 COSMIC genes were obtained due to a different DepMap Public version. Below are the resulting plots. MMLP clearly performs better than TGDRP in the LPO and LCO scenarios, but the RMSEs are very high (around 2), and considering that the drug response values are $\log_{10}$-transformed, this translates to predicted values that are roughly 100 ($\approx 10^2$) times the original drug response or about $\frac{1}{100}$ of it. In the LDO scenario, TGDRP performs even worse than the mean predictor.

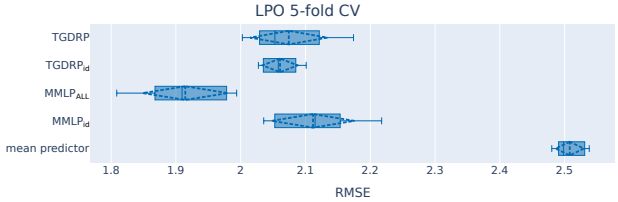


Figure 1: LPO 5-fold CV results for DepMap Public 24Q2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation.

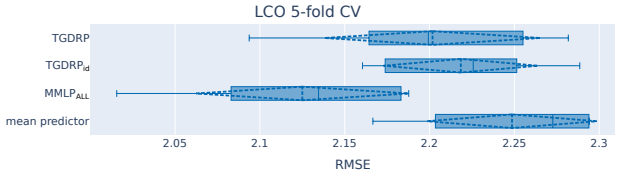


Figure 2: LCO 5-fold CV results for DepMap Public 24Q2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation.

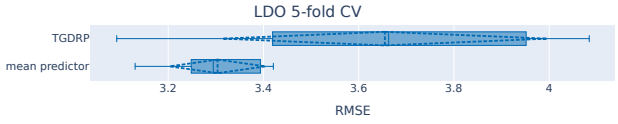


Figure 3: LDO 5-fold CV results for DepMap Public 24Q2. Each boxplot is computed from the five RMSEs. The additional dashed lines are mean and standard deviation.

3.3 VUScope: a mathematical model for evaluating image-based drug response measurements and predicting long-term incubation outcome

This section provides a brief background and outlines the motivation behind the third publication. The publication itself is included thereafter.

3.3.1 Background and motivation

The second publication highlighted the pressing need to reconsider the currently used drug response data before developing new predictive models. Among the biological input data types MUT, CNV, and EXP, only EXP exhibited a high correlation between Cell Model Passports and DepMap, indicating that EXP may still hold promise for modeling. As for the output data, the standard drug response metrics IC_{50} and AUC showed only moderate or even no correlation across replicate experiments within GDSC for identical cell line-drug pairs. Similarly, cross-dataset correlation between the IC_{50} and AUC values of GDSC and CCLE for same cell line-drug pairs was also moderate. The low correlations observed in the output data may also hinder models to effectively learn from the chemical input data such as molecular graphs. If the output data exhibited stronger correlations, patterns in the molecular graphs might more clearly align with the output values. These findings suggest that retraining existing models with alternative output data could enhance predictive performance while keeping EXP and molecular graphs as inputs.

There are several problems with the current output metrics IC_{50}, E_{max}, and AUC obtained from traditional HTS. First, for each drug, only few datapoints are available for fitting the 4PL curve. This is due to the fact that typically, a small number of concentrations is tested, e.g., DepMap uses 7 or 8 concentrations. Given the technical variability, outliers can significantly influence the model, leading to skewed IC_{50}, E_{max}, and AUC values. Second, IC_{50}, E_{max}, and AUC are influenced by the cell proliferation rate, which makes them incomparable between different cell lines [22]. A drug may appear to perform well on one cell line but only moderately on another, despite being equally effective on both, see Figure 3.1. Third, traditional HTS can misrepresent drug efficacy, as some drugs exert stronger effects only after a long incubation time. For instance, Drug 1 might show a lower AUC than Drug 2 at 72 hours, but this relationship could reverse at 120 hours, see Figure 3.2.

Live-cell imaging could provide a solution to these issues. First, by testing the same set of drug concentrations at multiple time points, live-cell imaging provides more data points and reduces the impact of outliers. Second, it enables direct observation of cell proliferation over time, allowing this factor to be accounted for in the analysis. Third, models fitted to live-cell imaging data allow for the extrapolation of drug responses over extended time periods, offering a more complete view of drug efficacy, particularly in cases where effects become apparent only after prolonged exposure.

Therefore, a dose-time-response model that extends the traditional 4PL curve for dose response with time as an additional dimension was introduced, along with a new dose-time-dependent drug response metric, the growth rate inhibition volume under the dose-time-response surface (GRIVUS).

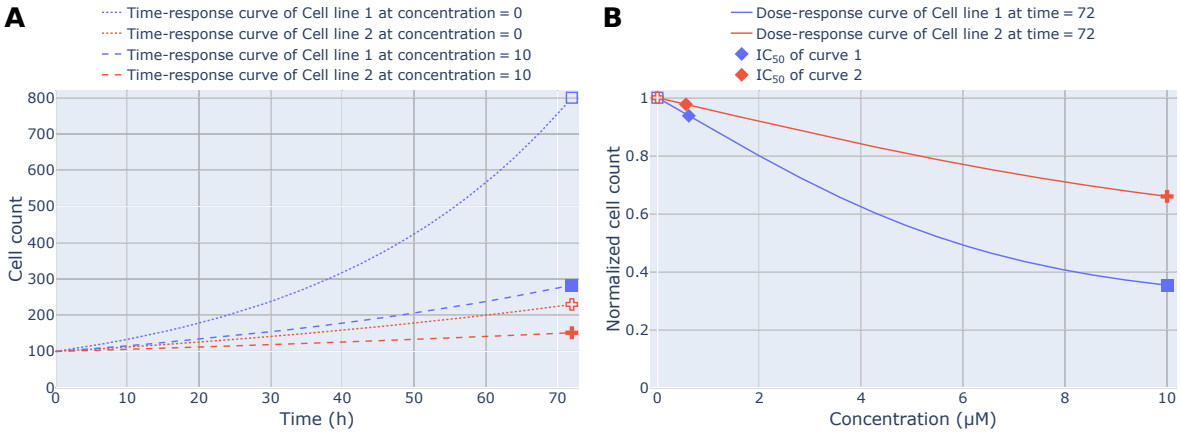


Figure 3.1: E_{max}, IC_{50}, and AUC cannot be directly compared across cell lines with different proliferation rates. Two different cell lines are either under negative control conditions (concentration = 0) or exposed to the same drug (concentration = 10). Panel A shows that Cell line 1 proliferates faster than Cell line 2 under negative control conditions, and that the drug halves proliferation rate in both. Panel B illustrates that despite identical drug effect, neither E_{max}, IC_{50}, nor AUC is identical across the cell lines. E_{max} and IC_{50} for Cell line 1 are higher, while the AUC is lower than for Cell line 2, suggesting higher, lower, or higher sensitivity, respectively. The marked data points (empty and filled squares and crosses) represent the same measurements in both panels, with normalization by the corresponding negative control cell count applied in Panel B.

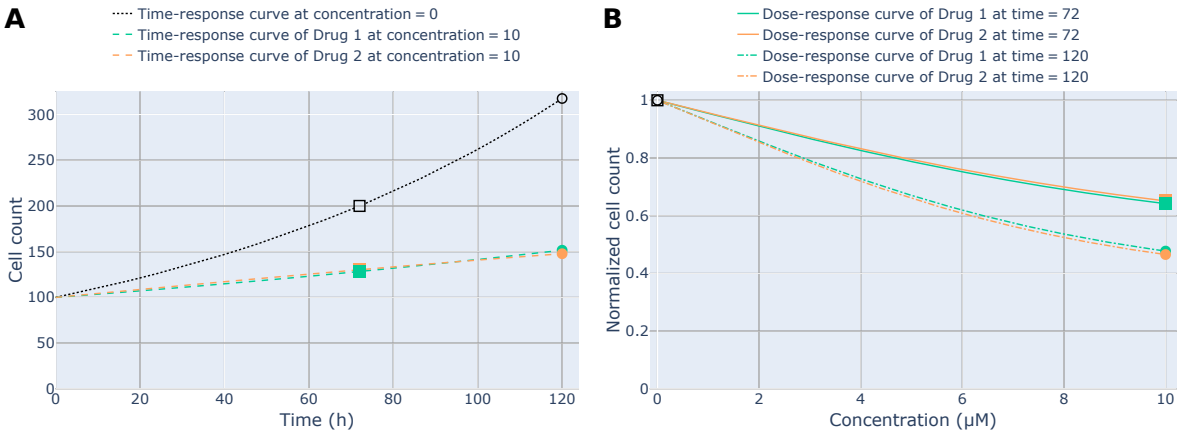


Figure 3.2: Longer incubation time can reveal delayed drug efficacy. The same cell line is either under negative control conditions (concentration = 0) or exposed to one of two drugs. In Panel A, the time-response curve of Drug 2 initially shows less growth inhibition than the curve of Drug 1, but eventually reaches a lower value owing to stronger saturation. As a result, Panel B shows Drug 1 with a lower AUC at 72 hours, implying higher efficacy; however, Drug 2 is more effective by 120 hours. The marked data points (empty and filled squares and circles) represent the same measurements in both panels, with normalization by the corresponding negative control cell count applied in Panel B.

3.3.2 Publication

Bioinformatics, 2026, **42(2)**, btaf679
https://doi.org/10.1093/bioinformatics/btaf679
Advance Access Publication Date: 4 February 2026
Original Paper

OXFORD

Systems biology

VUScope: a mathematical model for evaluating image-based drug response measurements and predicting long-term incubation outcomes

Nguyen Khoa Tran[1,2,†], My Ky Huynh[1,†], Alexander D. Kotman[3], Martin Jürgens[1,2], Thomas Kurz[4], Sascha Dietrich[3], Gunnar W. Klau[1,2,‡], Nan Qin[3,5,6,*,‡]

[1]Department of Computer Science, Heinrich Heine University Düsseldorf, Düsseldorf, 40225, Germany
[2]Center for Digital Medicine, Heinrich Heine University Düsseldorf, Düsseldorf, 40599, Germany
[3]Clinic of Hematology, Oncology, and Clinical Immunology, University Hospital of Düsseldorf, Düsseldorf, 40225, Germany
[4]Institute of Pharmaceutical and Medicinal Chemistry, Heinrich Heine University Düsseldorf, Düsseldorf, 40225, Germany
[5]Spatial & Functional Screening Core Facility, University Hospital of Düsseldorf, Düsseldorf, 40225, Germany
[6]Center for Integrated Oncology, Mildred Scheel School of Oncology Aachen-Bonn-Cologne-Düsseldorf, Düsseldorf, 40225, Germany

*Corresponding author. Spatial & Functional Screening Core Facility, University Hospital of Düsseldorf, Moorenstr. 5, Düsseldorf, 40225, Germany.
E-mail: nan.qin@med.uni-duesseldorf.de

[†]Shared first authors.

[‡]Shared last authors.

Associate Editor: Jianlin Cheng

Abstract

Motivation: Live-cell imaging-based drug screening increases the likelihood of identifying effective and safe drugs by providing dynamic, high-content, and physiologically relevant data. As a result, it improves the success rate of drug development and facilitates the translation of bench-side discoveries to bedside applications. Despite these advantages, no comprehensive metrics currently exist to evaluate dose–time-dependent drug responses. To address this gap, we established a systematic framework to assess drug effects across a range of concentrations and exposure durations simultaneously. This metric enables more accurate evaluation of drug responses measured by live-cell imaging.

Results: We employed treatment concentrations ranging from 0 to 10 μM and performed live-cell imaging-based measurements over a 120-h incubation period. To analyze the experimental data, we developed VUScope, a new mathematical model combining the 4-parameter logistic curve and a logistic function to characterize dose–time-dependent responses. This enabled us to calculate the Growth Rate Inhibition Volume Under the dose–time–response Surface (GRIVUS), which serves as a critical metric for assessing dynamic drug responses. Furthermore, our mathematical model allowed us to predict long-term treatment responses based on short-term drug responses. We validated the predictive capabilities of our model using independent datasets and observed that VUScope enhances prediction accuracy and offers deeper insights into drug effects than previously possible. By integrating VUScope into high-throughput drug screening platforms, we can further improve the efficacy of drug development and treatment selection.

Availability and implementation: We have made VUScope more accessible to users conducting pharmacological studies by uploading a detailed description, example datasets, and the source code to vuscope.albi.hhu.de, https://github.com/AlBi-HHU/VUScope, and https://doi.org/10.5281/zenodo.17610533.

1 Introduction

The selection of a 48–72-h incubation period in drug screening experiments is widely adopted as it effectively balances the need for sufficient time for the drug to influence cellular processes with the practicalities of experimental workflows. Specifically, a 72-h incubation period is generally adequate for capturing effects on most mammalian cells, which typically exhibit doubling times in the range of 20–40 h (He *et al.* 2022). Many assays, such as MTT and CellTiter-Glo (CTG), can more reliably detect drug effects after this elapsed time, ensuring high signal-to-noise ratios (Kleijn *et al.* 2016, Abebe *et al.* 2021). Nonetheless, some drugs may act rapidly or exhibit transient effects. An insufficient endpoint might overlook early events or fail to differentiate between immediate and delayed drug responses. Certain compounds, such as epigenetic inhibitors, may require a more extended incubation period to achieve optimal targeted effects (Bauden *et al.* 2015).

To overcome these limitations, employing label-free and noninvasive live-cell imaging techniques facilitates continuous observation of cellular processes over time. This approach enables real-time monitoring of the dynamic effects of drug candidates on living cells, offering insights often missed by traditional endpoint measurement approaches. Ultimately, it provides a more comprehensive understanding of the mechanisms and temporal effects of drugs. However, traditional metrics like IC50, EC50, E_{max}, or area under the dose–response curve (AUC), derived from endpoint assays, do not capture the temporal drug effects observed through live-cell imaging.

Received: 30 July 2025; **Revised:** 14 November 2025; **Accepted:** 17 December 2025

Consequently, in this study, we aimed to develop a model to calculate dose–time-dependent drug response metrics.

By combining the 4-parameter logistic curve and a logistic function to characterize dose–time-dependent responses, we delivered real-time assessments of drug efficacy and predicted long-term drug responses based on data obtained from shorter incubation periods. Our innovative model, VUScope, has the potential to transform imaging-based drug screening into a cornerstone of pharmacological research.

Hafner *et al.* (2016) introduced the growth rate inhibition (GR) metric to remove the impact of growth rate on drug response. They further extended their dose-dependent GR model by incorporating time as an additional variable, aiming to capture delayed effects, drug adaptation, variable kinetics of drug–target interactions, and drug efflux. However, as discussed in Supplementary Material A, available as supplementary data at *Bioinformatics* online, their GR metric itself is not time-dependent. To address this limitation, we propose a new dose–time-dependent metric derived from VUScope: Growth Rate Inhibition Volume Under the dose–time–response Surface (GRIVUS). To the best of our knowledge, GRIVUS is the first time-dependent drug response metric, enabling the evaluation of dynamic drug responses while reducing reliance on trial-and-error experiments, ultimately optimizing preclinical research. It extends the AUC metric by the additional time dimension and integrates the core idea of the GR metric.

2 Materials and methods

This section presents the materials and methods used. Key resources, i.e. experimental models and chemicals, are summarized in Table 1, available as supplementary data at *Bioinformatics* online. Data and code are available on vuscope.albi.hhu.de.

2.1 Cell culture

Cells were cultured following Good Cell Culture Practice guidelines. Generally, cells were maintained in Dulbecco's Modified Eagle's Medium (DMEM, Gibco, South America) supplemented with 10% Fetal Bovine Serum (FBS, Sigma). The cells were incubated at 37°C in a 5% CO_2 atmosphere and were passaged as needed when confluency reached approximately 80%. All experiments utilized mycoplasma-free and authenticated cell lines. Authentication was performed regularly using Short Tandem Repeat (STR) analysis by the Genomics & Transcriptomics Laboratory at the Biological and Medical Research Center (BMFZ) at Heinrich Heine University Düsseldorf, Germany.

2.2 Imaging

To prevent cell overgrowth at 120h, optimal seeding numbers for each cell line were carefully validated before starting live-cell imaging. Cells were distributed uniformly into 384-well plates using the Multidrop Combi (Thermo Fisher Scientific, Waltham, USA). Cell counts were measured at time zero and again at 3 h, with wells showing major deviations excluded from analysis. Each plate had six healthy control wells to monitor untreated cells. Imaging was carried out every 3 h over 120 h using an Incucyte SX5 (Sartorius, Göttingen, Germany, 10× objective) while maintaining 37°C and 5% CO_2. Images were analyzed by Incucyte's Cell-by-Cell module, with resulting counts exported for further analysis and normalized to initial counts of 1 for each cell line–drug pair.

2.3 CellTiter-Glo (CTG) luminescent cell viability assay

The CTG reagent (Promega, MA, USA) was prepared following the manufacturer's instructions for the cell viability assessment. To maintain exponential growth throughout the experiment, the optimal cell concentrations were determined experimentally. Cells were seeded in 384-well plates at a volume of 30 μl per well. After incubation, the CTG reagent was added to lyse the cells, and luminescence was measured using a Spark 10 M microplate reader (Tecan, Männedorf, Switzerland).

2.4 Inhibitor libraries and drug screening

Drug screening was conducted at the Spatial & Functional Core Facility of the Medical Faculty at Heinrich Heine University Düsseldorf. Sample preparation and data processing were carried out as previously described (Jeising *et al.* 2024). A library of 18 histone deacetylase inhibitors was used to profile the drug response in four cell lines. All compounds were tested at concentrations ranging from 0 to 10 μM, covering six distinct concentration levels.

2.5 Statistical metrics

The mean absolute percentage error (MAPE) is the relative deviation between actual values A_t and forecast values F_t, and is defined as $\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$.

In contrast to the Pearson correlation coefficient (PCC), the concordance correlation coefficient (CCC) is a measure not only for correlation, but also for agreement and concordance, and is defined as $\text{CCC} = \frac{2\text{PCC}\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2+(\mu_x-\mu_y)^2}$, where x and y represent two variables, μ_x and μ_y are their means, and σ_x^2 and σ_y^2 are their variances. The CCC is commonly reported along with a 95% confidence interval (CI) derived from bootstrap analysis with 100 repetitions.

2.6 VUScope: a dose–time–response model

Dose–response relationships are conventionally characterized by the 4-parameter logistic model. To capture time-dependent responses, we incorporate a logistic function to model cell proliferation over time. Our innovative approach combines time dependence with dose dependence, which is mathematically expressed as follows:

$$f(d,t) = \frac{\alpha(t) - \delta(t)}{1 + 10^{\beta(t)\cdot(d-\gamma(t))}} + \delta(t).$$

Here, $\alpha(t) = \frac{a_\alpha \cdot 2^{k_\alpha \cdot t}}{(a_\alpha - 1) + 2^{k_\alpha \cdot t}}$ with growth rate $k_\alpha \in \mathbb{R}$ and upper asymptote $a_\alpha \in \mathbb{R}$ represents the cell count at a time t in the absence of drug application. The term $\delta(t) = \frac{a_\delta \cdot 2^{k_\delta \cdot t}}{(a_\delta - 1) + 2^{k_\delta \cdot t}}$ with growth rate $k_\delta \in \mathbb{R}$ and upper asymptote $a_\delta \in \mathbb{R}$ represents the cell count at time t when subjected to an infinitely high drug dosage. Further, $\beta(t) = k_\beta \cdot |\alpha(t) - \delta(t)|$ with scaling factor $k_\beta \in \mathbb{R}$, $k_\beta \geq 0$, reflects the steepness of the curve surrounding $\gamma(t)$ at time t. Finally, $\gamma(t) = k_\gamma$ with $k_\gamma \in \mathbb{R}$ represents the $\log_{10}(\text{IC50})$ (or $\log_{10}(\text{EC50})$ for growth-stimulating drugs). The parameters $k_\alpha, a_\alpha, k_\delta, a_\delta, k_\beta, k_\gamma$ are estimated using the `least_squares` function in SciPy (Virtanen *et al.* 2020), which we configured to minimize the

MAPE. The modeling choices for the time-dependent parameter functions are discussed in Supplementary Material F, available as supplementary data at *Bioinformatics* online.

2.7 GRIVUS

After fitting a dose–time–response surface to time-course data, drug response can be quantified with metrics such as the Volume Under the dose–time–response Surface (VUS), a 3D extension of the traditional AUC. Analogous to the GR metric (Hafner *et al.* 2016), we refined the VUS by removing the impact of growth rate to develop the GRIVUS. This refinement involved expressing both the growth rate of the control condition (k_α) and that of the treated condition (k_δ) as a ratio to the control growth rate (k_α). In cases where $k_\alpha < k_\delta$, we expressed both values as a ratio to k_δ instead of k_α, ensuring the exponential prefactors of t remain below 1 to prevent numerical instability. This results in $\alpha'(t) = \frac{a_\alpha \cdot 2^{\frac{k_\alpha}{\max\{k_\alpha,k_\delta\}}t}}{(a_\alpha - 1) + 2^{\frac{k_\alpha}{\max\{k_\alpha,k_\delta\}}t}}$, $\delta'(t) = \frac{a_\delta \cdot 2^{\frac{k_\delta}{\max\{k_\alpha,k_\delta\}}t}}{(a_\delta - 1) + 2^{\frac{k_\delta}{\max\{k_\alpha,k_\delta\}}t}}$, and $\beta'(t) = k_\beta \cdot |\alpha'(t) - \delta'(t)|$ such that GRIVUS is calculated not from the original function $f(d,t)$, but from

$$f'(d,t) = \frac{\alpha'(t) - \delta'(t)}{1 + 10^{\beta'(t)\cdot(d - \gamma(t))}} + \delta'(t).$$

GRIVUS is computed using numerical approximation (Fig. 1B). For data processing, we implemented normalization to minimize bias in the results caused by larger-scale values. Initially, we computed the unaffected GRIVUS value under the assumption that cell proliferation was not influenced by the administered drug at the minimum concentration. We then normalized the GRIVUS value of $f'(d,t)$ by dividing it by the unaffected GRIVUS value (Fig. 1C).

3 Results

The following section presents results from model fitting as well as extrapolation to predict later time points.

3.1 VUScope accurately fits data from live-cell imaging experiments, even with the imaging conducted at 24-h intervals

We used VUScope to accurately fit data from a live-cell imaging experiment, with images acquired every 3 h (Fig. 1A). The resulting 3D-fitted model was quantified by calculating normalized GRIVUS values, as described in Section 2, with calculation strategies illustrated in Fig. 1B and C. To evaluate the accuracy of the VUScope dose–time–response surface, we compared fitted and actual data points using MAPE, a scale-independent metric. Across all cell line–drug pairs, the average MAPE was 8.30%±3.16, highlighting VUScope's effectiveness for analyzing drug responses from live-cell imaging data. We repeated this analysis with images taken at 24-h intervals (Fig. 1D), observing a slightly higher average MAPE of 8.71%±3.43. Additionally, GRIVUS values for all cell line–drug pairs at both intervals were compared using the

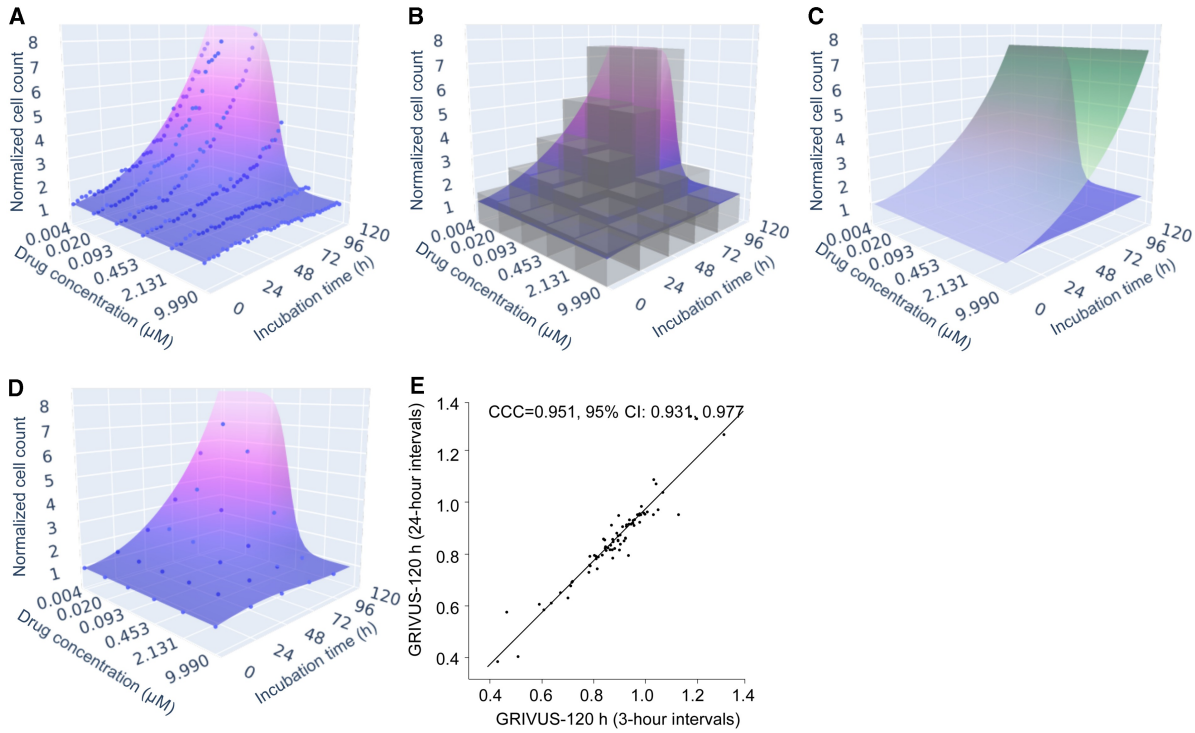


Figure 1. VUScope effectively captures data from imaging-based experiments. (A) 3D fitting of VUScope. The model was constructed using the 4-parameter logistic model in conjunction with a logistic function to represent time-dependent cell growth. (B) The Growth Rate Inhibition Volume Under the dose–time–response Surface (GRIVUS) is calculated using numerical approximation. While the actual number of cuboids is 10 000, this was reduced to 25 for purposes of visualization. Also, we approximate the GRIVUS from above (as seen in the figure) and below, and average both values. (C) The normalized GRIVUS value is calculated by dividing the measured GRIVUS value (blue) by the unaffected GRIVUS value (green). (D) VUScope successfully fits the imaging data using images taken at 24-h intervals. (E) The CCC was computed to compare the GRIVUS values of all cell line–drug pairs calculated from images acquired at 3-h intervals with those from 24-h intervals.

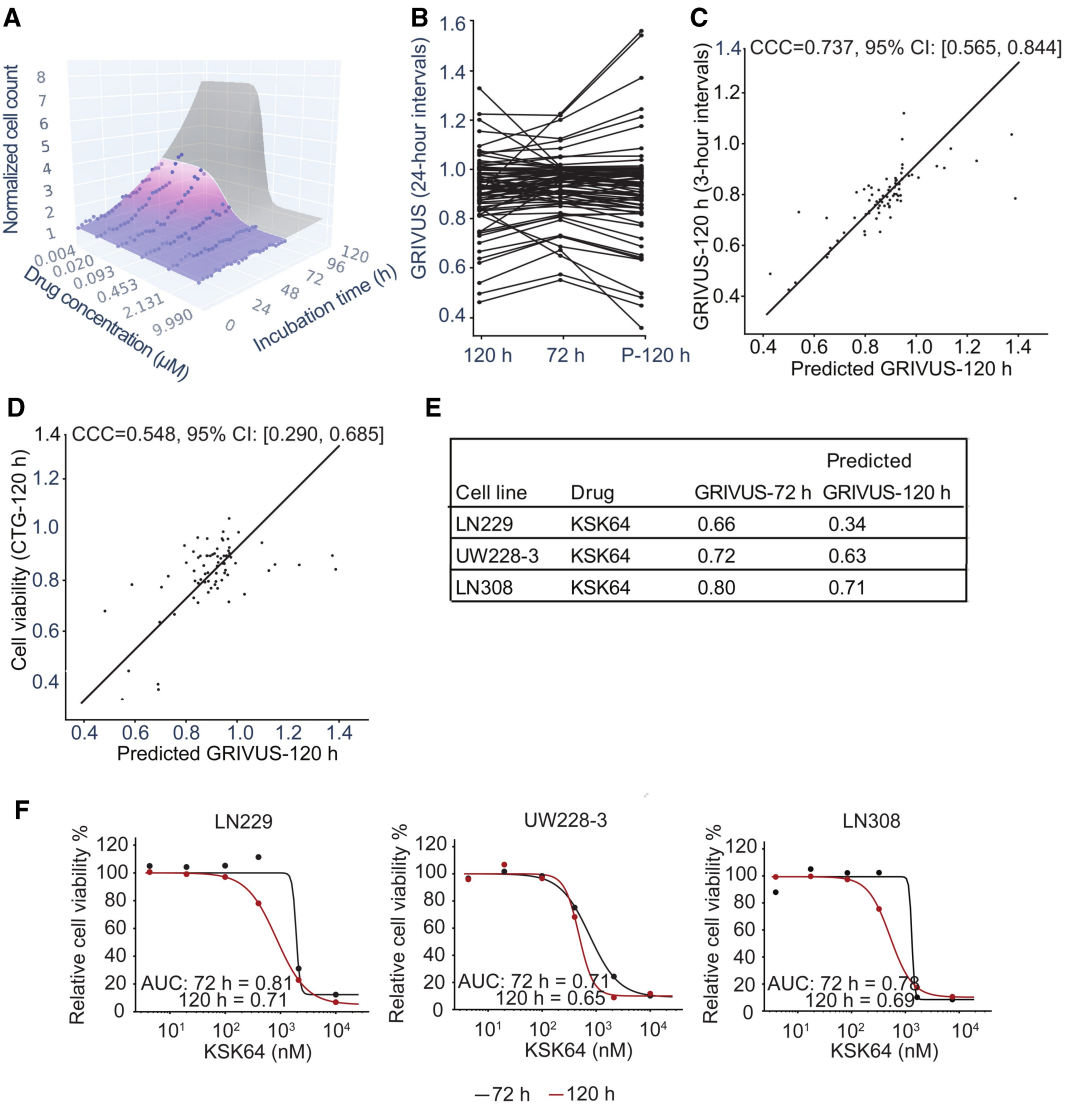
CCC. Figure 1E demonstrates nearly perfect concordance, emphasizing the robustness of our fitting model even with reduced data frequency.

3.2 VUScope can effectively predict drug effects from short-term treatment data for long-term therapy

We evaluated the predictive capability of VUScope by first fitting the dose–time–response model to data collected over the initial 72 h of incubation, using images taken at either 3-h or 24-h intervals. Across all cell line–drug pairs, the average MAPE was 6.14%±2.37 for 3-h intervals and 6.78%±2.79 for 24-h intervals. After calculating the GRIVUS at 72 h, we extracted the optimized parameters and used them to extrapolate the model for predicting GRIVUS at 120 h (Fig. 2A). The model's 120-h predictions were then assessed against the 120-h fit from the previous subsection. The average MAPE rose only slightly from 8.30%±3.16 to 9.43%±3.80 (3-h intervals) and 9.28%±3.47 (24-h intervals) at 120 h, indicating robust model performance.

In addition, we employed the VUScope fit to the complete 120-h experimental data to calculate the GRIVUS values, which we refer to as *measured* GRIVUS values in the following.

Analysis revealed that, when comparing changes in GRIVUS after 72 h of incubation with the predicted and measured GRIVUS values at 120 h, we observed that the trend, whether there was an increase or decrease in GRIVUS following the 48-h extended incubation, remained consistent across most cell line–drug pairs (Fig. 2B). There was strong



Cell line	Drug	GRIVUS-72 h	Predicted GRIVUS-120 h
LN229	KSK64	0.66	0.34
UW228-3	KSK64	0.72	0.63
LN308	KSK64	0.80	0.71

Figure 2. VUScope accurately predicts drug response by analyzing images captured every 3 h. (A) Data from the first 72 h of treatment (blue) were 3D fitted to predict responses after an additional 48 h of incubation (gray). (B) Changes in GRIVUS values observed at 72 h are compared to both the measured and predicted GRIVUS values at 120 h (P-120 h) through paired analysis. (C) Concordance correlation coefficient (CCC) analysis compares the predicted and measured GRIVUS values across all cell line–drug pairs. (D) CCC is also used to compare predicted GRIVUS values at 120 h with cell viability measured by the CTG assay after 120 h of treatment. (E) For KSK64, summarized outcomes are presented using either GRIVUS values at 72 h or predicted GRIVUS at 120 h, demonstrating a larger decline in GRIVUS and suggesting a more specific effect at 120 h. (F) The dose–response curve illustrates the endpoint measurement obtained through the CTG assay after two different treatment durations: 72 h (black) and 120 h (red). The reduction in area under the dose–response curve (AUC) values observed after the 120-h treatment, as compared to the 72-h treatment, underscores the more effective targeted effect of KSK64 following the extended treatment duration.

concordance between measured and predicted GRIVUS at 120 h as well (Fig. 2C). To independently validate VUScope's predictive accuracy, we conducted ATP-based cell viability assays at 72 and 120 h of incubation and compared those results with GRIVUS predictions. Despite the methodological differences and resulting variability between ATP-based and live-cell imaging measurements, the observed correlation was strong (Fig. 2D), showing that our model accurately reflects the temporal drug response.

Moreover, we specifically evaluated the targeted effect of KSK64, a newly developed HDAC inhibitor (Friedrich *et al.* 2020), through ATP-based endpoint measurements after either 72 or 120 h of treatment. The resulting dose–response curves corroborated the findings from VUScope (Fig. 2E) and demonstrated an increased targeted effect of KSK64, indicated by lower AUC and GRIVUS values, in three tested cell lines after 120 h of treatment compared to 72 h (Fig. 2F).

Complete results and figures for the data using 24-h intervals are presented in Fig. 1, available as supplementary data at *Bioinformatics* online. Additionally, we investigated VUScope's ability to predict drug responses over longer incubation times. By analyzing only the first 48 h of data, we predicted responses at 120 h. Despite the challenge of limited data points affecting model fitting, VUScope's predictions significantly correlated with actual outcomes. Importantly, the predicted changes in drug sensitivity matched the observed directions (Fig. 2, available as supplementary data at *Bioinformatics* online). These results demonstrate that VUScope can accurately forecast long-term drug responses from short-term measurements.

4 Discussion

Traditional drug response metrics like IC50, EC50, E_{max}, and AUC capture effects at a single time point, reducing complex temporal drug behaviors to a single value. This simplification often misses nuanced responses such as delayed onset or cumulative effects. These metrics are further influenced by differences between cell types; e.g. fast- and slow-growing cells can respond differently due to their inherent properties rather than the drug itself. Variations in culture conditions and initial seeding density also impact drug sensitivity measurements (Haverty *et al.* 2016).

To address variability due to growth rates and experimental noise, the normalized drug response metric incorporates starting and ending data for treatments (Gupta *et al.* 2020); however, it still overlooks temporal response dynamics. GRIVUS addresses this by combining GR value-based proliferation adjustments with continuous, time-resolved monitoring. This approach enables better characterization of response kinetics, resistance patterns, growth variability, and the distinction between cumulative and transient effects. As a result, using GRIVUS for live-cell or fresh tissue imaging time courses offers deeper insight into dynamic biological processes and enables more comprehensive analyses, often with less sample material (Angeli *et al.* 2024).

Most large-scale studies of cellular response to anti-cancer drugs measure changes in IC50, EC50, E_{max}, or AUC (Barretina *et al.* 2012, Yang *et al.* 2013, Corsello *et al.* 2020). However, when concentration points are insufficient, curve fitting is less reliable, increasing errors in these estimates. Noisy measurements at each dose further increase uncertainty, making Hill slopes and plateaus difficult to estimate and resulting in shifting or imprecise IC50, EC50, E_{max}, and AUC values (Hafner *et al.* 2017). For instance, outliers can greatly influence AUC values (Fig. 3A, available as supplementary data at *Bioinformatics* online), making the 120-h treatment appear more effective than the 72-h one. In contrast, GRIVUS displays high robustness to strong outliers (Fig. 3B, available as supplementary data at *Bioinformatics* online) and still yields accurate predictions (Fig. 3C, available as supplementary data at *Bioinformatics* online).

While prior reports often focus on single-concentration, time-dependent responses (Angeli *et al.* 2024, Colling *et al.* 2024) or on simulated dose–time–response models lacking lab validation (Jackson and Byrne 2000, Murphy *et al.* 2020), our study uniquely integrates substantial experimental (wet lab) data to support model training and utilizes an independent approach to confirm our image-based analyses.

4.1 Limitations and future work

Our study deliberately used newly developed HDAC inhibitors, which modify gene expression by altering chromatin structure (Bolden *et al.* 2006). This process may require extended exposure time to accurately assess their targeted effects. Through our dose–time-dependent measurements, incorporating both predicted and measured GRIVUS values, we clearly demonstrated that certain HDAC inhibitors require more than 72 h to achieve their intended effects. This finding indicates that existing studies on HDAC inhibitors that utilize a 72-h treatment duration may underreport their targeted impacts (Ecker *et al.* 2021, Marquardt *et al.* 2023). It also highlights the importance of employing our VUScope and GRIVUS value to determine the appropriate treatment duration rather than relying on a standard 72-h protocol for all cases.

While our model provides valuable insights, it also has limitations that we need to address. Our study focused on a selected range of cell lines and examined the dose–time-dependent drug response to HDAC inhibitors, a specific class of drugs. This focus may influence the robustness and broader applicability of our model. To enhance its effectiveness, we will develop an online tool to assist users in evaluating their live-cell imaging data and optimizing our model simultaneously. By constructing a comprehensive database with adequate training datasets, we can improve the diversity, completeness, and reproducibility of our model.

Currently, our primary focus is on cell count changes, but imaging can yield many additional insights that are worth integrating. Moving forward, it is essential to develop a comprehensive imaging analysis tool that captures data on cell morphology and behavior, growth characteristics in both mono- and co-cultures, and various forms of cell death, such as apoptosis, necrosis, and ferroptosis. Excluding dead cells from cell counts can cause normalized values to drop below 1 and approach 0. In theory, VUScope could already capture this phenomenon, as its growth rates k_α and k_δ can take negative values; however, this still needs to be validated with appropriate datasets.

By combining this approach with functional fluorescent reporter assays, we can delve deeper into intracellular organelle dynamics, subcellular protein distribution, activation of signaling pathways, and ion channel activity. This integrated method would lead to a new era of "high-definition" drug response profiling, providing unique insights into therapeutic

mechanisms of action and enabling robust quantification of drug responses.

Acknowledgements

We would like to thank Sartorius for their excellent technical support.

Author contributions

Nguyen Khoa Tran (Conceptualization [lead], Data curation [lead], Formal analysis [lead], Methodology [lead], Software [lead], Supervision [lead], Validation [lead], Visualization [lead], Writing—original draft [supporting], Writing—review & editing [supporting]), My Ky Huynh (Data curation [lead], Formal analysis [lead], Methodology [lead], Software [lead], Validation [lead]), Alexander D. Kotman (Data curation [lead]), Martin Jürgens (Data curation [supporting], Formal analysis [supporting], Methodology [supporting]), Thomas Kurz (Data curation [supporting], Methodology [supporting]), Sascha Dietrich (Data curation [supporting], Formal analysis [supporting], Funding acquisition [supporting], Investigation [supporting], Methodology [supporting], Project administration [supporting], Resources [supporting], Software [supporting], Supervision [supporting]), Gunnar W. Klau (Conceptualization [lead], Data curation [supporting], Formal analysis [lead], Funding acquisition [supporting], Investigation [supporting], Methodology [lead], Project administration [lead], Resources [lead], Software [lead], Supervision [lead], Validation [supporting], Visualization [supporting], Writing—original draft [supporting], Writing—review & editing [supporting]), and Nan Qin (Conceptualization [lead], Data curation [supporting], Funding acquisition [lead], Methodology [supporting], Project administration [lead], Supervision [lead], Visualization [lead], Writing—original draft [lead], Writing—review & editing [lead])

Supplementary data

Supplementary data is available at *Bioinformatics* online.

Conflict of interest: No competing interest is declared.

Funding

This research was funded by the German Childhood Cancer Foundation (Bonn, NRW, Germany), grant count DKS 2021.20, Stiftung Kinderkrebsklinik (Düsseldorf, NRW, Germany), grant count 3267, and Research Committee of the Medical Faculty (Düsseldorf, NRW, Germany), grant counts FOKO 2021-44 and FOKO 2024-64.

Data availability

All data related to this manuscript are available and described either within the manuscript or through the URL (vuscope.albi.hhu.de, https://github.com/AlBi-HHU/VUScope, and https://doi.org/10.5281/zenodo.17610533).

References

Abebe F, Hopkins M, Vodnala S *et al.* Development of a rapid in vitro screening assay using metabolic inhibitors to detect highly selective anticancer agents. *ACS Omega* 2021;**6**:18333–43.

Angeli C, Wroblewska J, Klein E *et al.* Protocol to generate scaffold-free, multicomponent 3D melanoma spheroid models for preclinical drug testing. *STAR Protoc* 2024;**5**:103058.

Barretina J, Caponigro G, Stransky N *et al.* The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* 2012;**483**:603–7.

Bauden M, Tassidis H, Ansari D. In vitro cytotoxicity evaluation of HDAC inhibitor Apicidin in pancreatic carcinoma cells subsequent time and dose dependent treatment. *Toxicol Lett* 2015;**236**:8–15.

Bolden J, Peart M, Johnstone R. Anticancer activities of histone deacetylase inhibitors. *Nat Rev Drug Discov* 2006;**5**:769–84.

Colling K, Symons E, Buroni L *et al.* Multiplexed live-cell imaging for drug responses in patient-derived organoid models of cancer. *JoVE* 2024. https://doi.org/10.3791/66072

Corsello S, Nagari R, Spangler R *et al.* Discovering the anticancer potential of non-oncology drugs by systematic viability profiling. *Nat Cancer* 2020;**1**:235–48.

Ecker J, Thatikonda V, Sigismondo G *et al.* Reduced chromatin binding of MYC is a key effect of HDAC inhibition in MYC amplified medulloblastoma. *Neuro Oncol* 2021;**23**:226–39.

Friedrich A, Assmann A, Schumacher L *et al.* In vitro assessment of the genotoxic hazard of novel hydroxamic acid-and benzamide-type histone deacetylase inhibitors (HDACi). *Int J Mol Sci* 2020;**21**:4747.

Gupta A, Gautam P, Wennerberg K *et al.* A normalized drug response metric improves accuracy and consistency of anticancer drug sensitivity quantification in cell-based screening. *Commun Biol* 2020;**3**:42.

Hafner M, Niepel M, Chung M *et al.* Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs. *Nat Methods* 2016;**13**:521–7.

Hafner M, Niepel M, Subramanian K *et al.* Designing drug–response experiments and quantifying their results. *Curr Protoc Chem Biol* 2017;**9**:96–116.

Haverty P, Lin E, Tan J *et al.* Reproducible pharmacogenomic profiling of cancer cell line panels. *Nature* 2016;**533**:333–7.

He Q, Rehmann M, Tian J *et al.* Improved titer in late-stage mammalian cell culture manufacturing by re-cloning. *Bioengineering* 2022;**9**:173.

Jackson T, Byrne H. A mathematical model to study the effects of drug resistance and vasculature on the response of solid tumors to chemotherapy. *Math Biosci* 2000;**164**:17–38.

Jeising S, Nickel A, Trübel J *et al.* A clinically compatible in vitro drug-screening platform identifies therapeutic vulnerabilities in primary cultures of brain metastases. *J Neurooncol* 2024;**169**:613–23.

Kleijn A, Kloezeman J, Balvers R *et al.* A systematic comparison identifies an ATP-based viability assay as most suitable read-out for drug screening in glioma stem-like cells. *Stem Cells Int* 2016;**2016**:5623235.

Marquardt V, Theruvath J, Pauck D *et al.* Tacedinaline (CI-994), a class I HDAC inhibitor, targets intrinsic tumor growth and leptomeningeal dissemination in MYC-driven medulloblastoma while making them susceptible to anti-CD47-induced macrophage phagocytosis via NF-kB-TGM2 driven tumor inflammation. *J Immunother Cancer* 2023;**11**:e005871.

Murphy H, McCarthy G, Dobrovolny H. Understanding the effect of measurement time on drug characterization. *PLoS One* 2020;**15**:e0233031.

Virtanen P, Gommers R, Oliphant T *et al.*; SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat Methods* 2020;**17**:261–72.

Yang W, Soares J, Greninger P *et al.* Genomics of drug sensitivity in cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res* 2013;**41**:D955–61.

Bioinformatics, 2026, 42, 1–6
https://doi.org/10.1093/bioinformatics/btaf679
Original Paper

A GR metrics are not time-dependent

Hafner et al. [6] state that

$$\mathrm{GR}(c,t) = 2^{\left(\frac{\log_2(x(c,t)/x_0)}{\log_2(x_{\mathrm{ctrl}}/x_0)}\right)} - 1 = 2^{\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}-\frac{1}{k}\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)} - 1,$$

with $x_{\mathrm{ctrl}} = x(0,t)$ and x_0 being the initial cell number, is independent of t. (The meanings of h, S_M, SC_{50}, LC_{50}, and k_L are not relevant in this context.) Hence, $\mathrm{GR}(c,t)$ can be written as $\mathrm{GR}(c)$.
The same goes for the time-dependent GR values Hafner et al. introduce: The authors state that GR values can be evaluated over a time interval $(2\times\Delta t)$ around any time point t based on the equation:

$$\mathrm{GR}(c,t) = 2^{\frac{\log_2(x(c,t+\Delta t)/x(c,t-\Delta t))}{\log_2(x(0,t+\Delta t)/x(0,t-\Delta t))}} - 1$$

However, substituting $x(c,t) = x_0\times\exp\left(t\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-t\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)$ yields

$$\begin{aligned}
\mathrm{GR}(c,t) &= 2^{\frac{\log_2\left(\left(x_0\times\exp\left((t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)/\left(x_0\times\exp\left((t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)\right)}{\log_2\left(\left(x_0\times\exp\left((t+\Delta t)\times k\left(1-\frac{S_M\times 0^h}{\mathrm{SC}_{50}^h+0^h}\right)-(t+\Delta t)\frac{k_L\times 0^h}{\mathrm{LC}_{50}^h+0^h}\right)\right)/\left(x_0\times\exp\left((t-\Delta t)\times k\left(1-\frac{S_M\times 0^h}{\mathrm{SC}_{50}^h+0^h}\right)-(t-\Delta t)\frac{k_L\times 0^h}{\mathrm{LC}_{50}^h+0^h}\right)\right)\right)}} - 1 \\
&= 2^{\frac{\log_2\left(\left(\exp\left((t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)/\left(\exp\left((t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)\right)}{\log_2((\exp((t+\Delta t)\times k))/(\exp((t-\Delta t)\times k)))}} - 1 \\
&= 2^{\frac{\log_2\left(\left(\exp\left((t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)/\left(\exp\left((t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)\right)}{\frac{1}{\ln(2)}\times(t+\Delta t)\times k-\frac{1}{\ln(2)}\times(t-\Delta t)\times k}} - 1 \\
&= 2^{\frac{\log_2\left(\left(\exp\left((t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)/\left(\exp\left((t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)\right)\right)}{\frac{1}{\ln(2)}\times 2\times\Delta t\times k}} - 1 \\
&= 2^{\frac{\frac{1}{\ln(2)}\times\left((t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)-\frac{1}{\ln(2)}\times\left((t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}\right)}{\frac{1}{\ln(2)}\times 2\times\Delta t\times k}} - 1 \\
&= 2^{\frac{(t+\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t+\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}-(t-\Delta t)\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-(t-\Delta t)\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}}{2\times\Delta t\times k}} - 1 \\
&= 2^{\frac{2\times\Delta t\times k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-2\times\Delta t\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}}{2\times\Delta t\times k}} - 1 \\
&= 2^{\frac{k\left(1-\frac{S_M\times c^h}{\mathrm{SC}_{50}^h+c^h}\right)-\frac{k_L\times c^h}{\mathrm{LC}_{50}^h+c^h}}{k}} - 1,
\end{aligned}$$

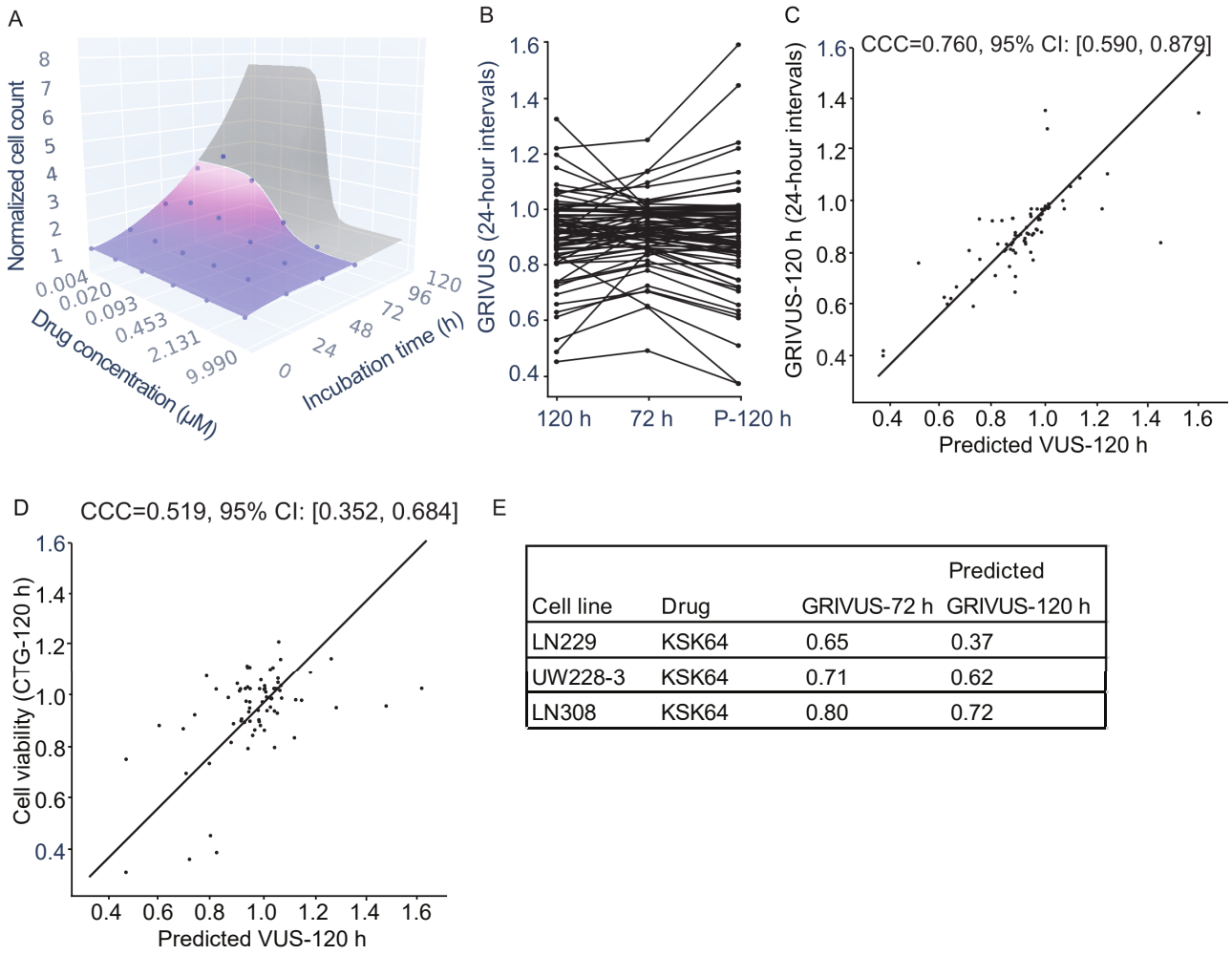
but this expression is independent of both t and Δt, thus none of the GR metrics $GR(c,t)$ are time-dependent.

B Key resources

Reagent or Resource	Source	Identifier
Experimental Models		
UW228-3	Gift from Dr. Landgraf, University Hospital of Düsseldorf (received: 2015)	RRID: CVCL_0573
LN229	Gift from Dr. Reifenberger, University Hospital of Düsseldorf (received: 2015)	RRID: CVCL_0393
T98G	Gift from Dr. Reifenberger, University Hospital of Düsseldorf (received: 2015)	RRID: CVCL_0556
LN308	Gift from Dr. Reifenberger, University Hospital of Düsseldorf (received: 2015)	RRID: CVCL_0934
All experiments were performed using authenticated cell lines free from mycoplasma contamination. Mycoplasma detection and authentication were performed every six months using the multiplex human cell line authentication test by Multiplexion (Heidelberg, Germany).		
Reagents		
DMEM	Thermo Fisher Scientific	12491023
FBS	Thermo Fisher Scientific	16140071
PBS	Thermo Fisher Scientific	10010023
Dimethylsulfoxid	PanReac AppliChem	A3672,0100
HDAC inhibitors		
Name	Molecular weight (g/mol)	Source
KSK64	406.44	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
HLK84	333.82	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
HLK54	301.35	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
HLK40	412.87	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
HLK38	397.86	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
LAK402	494.99	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
FJKK103	506.96	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
FJKK133	489.96	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
SHOK75	421.52	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
HLK89	504.41	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
FJKK94	562.04	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
FJKK81	598.51	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
MPK169	296.32	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
YAK376	423.40	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
YAK169	439.39	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)
FFK24	356.35	Gift from Dr. Kurz, Heinrich Heine University (received: 2022)

Table S1. Summary of experimental models and chemicals.

C Results for predictions with 24-hour interval data



Cell line	Drug	GRIVUS-72 h	Predicted GRIVUS-120 h
LN229	KSK64	0.65	0.37
UW228-3	KSK64	0.71	0.62
LN308	KSK64	0.80	0.72

Fig. S1. VUScope accurately predicts drug response by analyzing images captured every 24 hours. (A) Data from the first 72 hours of treatment (blue) were 3D-fitted to predict responses after an additional 48 hours of incubation (gray). (B) Changes in GRIVUS values observed at 72 hours are compared to both the measured and predicted GRIVUS values at 120 hours (P-120 h) through paired analysis. (C) Concordance correlation coefficient (CCC) analysis compares the predicted and measured GRIVUS values across all cell line-drug pairs. (D) CCC is also used to compare predicted GRIVUS values at 120 hours with cell viability measured by the CTG assay after 120 hours of treatment. (E) For KSK64, summarized outcomes are presented using either GRIVUS values at 72 hours or predicted GRIVUS at 120 hours, demonstrating a larger decline in GRIVUS and suggesting a more specific effect at 120 hours.

D Results for predictions with 24-hour interval data

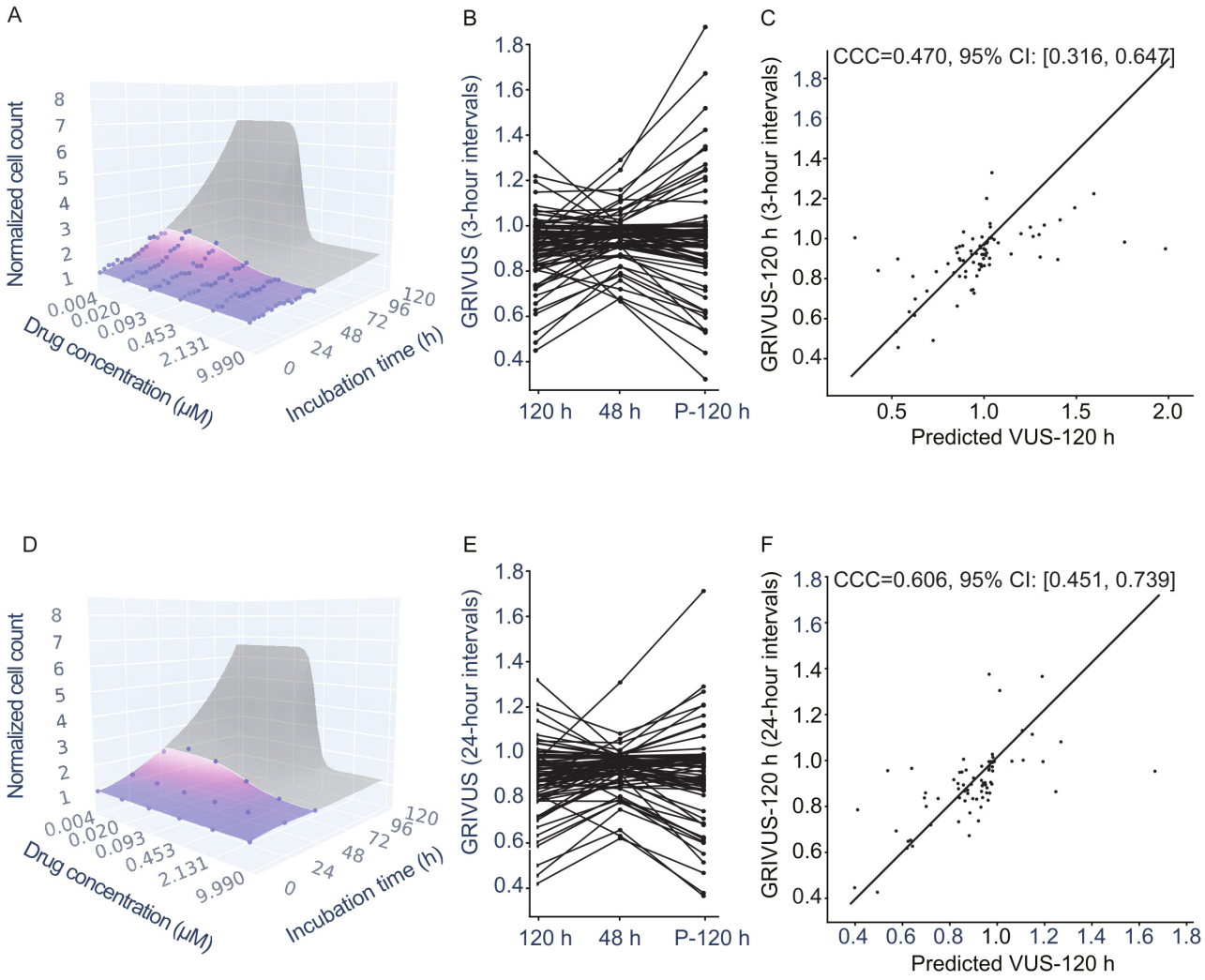


Fig. S2. VUScope predicts drug response by analyzing images during the initial 48 hours of incubation. (A) Images captured at 3-hour intervals in the first 48 hours (blue) were used to build 3D models predicting responses after another 72 hours (gray). (B) Changes in GRIVUS values at 48 hours, calculated from 3-hour interval images, were compared to both measured and predicted GRIVUS values at 120 hours (P-120 h) using paired analysis. (C) Concordance correlation coefficient (CCC) values, based on 3-hour interval images, assess agreement between predicted and measured GRIVUS values across all cell line-drug combinations. (D) Similarly, data from 24-hour interval images in the first 48 hours (blue) predicted response at 120 hours (gray). (E) GRIVUS value changes at 48 hours from 24-hour interval images were compared to measured and predicted values at 120 hours (paired analysis). (F) CCC values from 24-hour interval images compare predicted and measured GRIVUS values for all cell line-drug pairs.

E Outlier susceptibility of IC50, $E_{\max}$, AUC, and GRIVUS

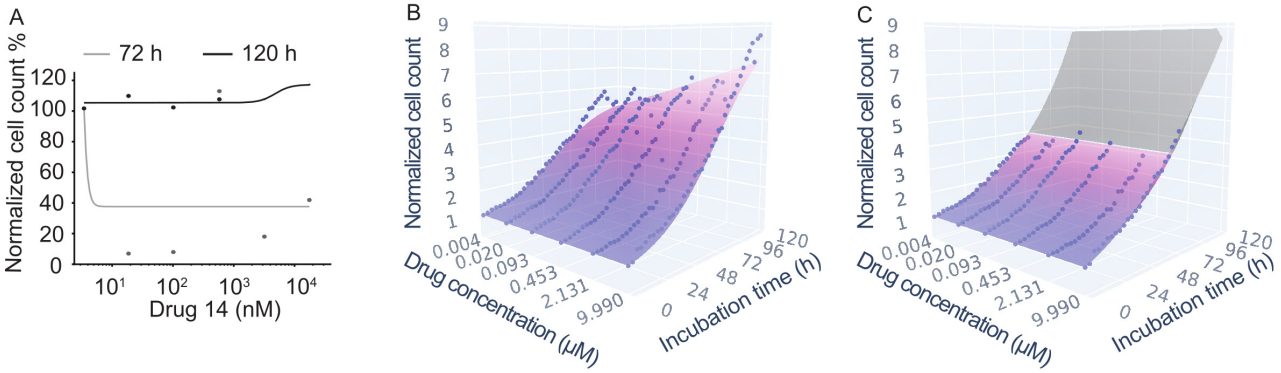


Fig. S3. Outliers can significantly skew IC50, $E_{\max}$, and AUC calculations, while GRIVUS values remain stable. (A) A sample dose-response curve demonstrates that introducing an outlier at 72 hours substantially impairs curve fitting, resulting in unreliable values for IC50, $E_{\max}$, and AUC. This outlier creates a marked difference between 72-hour and 120-hour results: AUC changes from 0.57 to 1.01, IC50 from −2.43 to 0.49, and $E_{\max}$ from 0.43 to −0.10. In contrast, GRIVUS remains robust. Even with several outliers at 120 hours (B), GRIVUS calculations are consistent, with 72-hour (C) and 120-hour values being similar (1.07 and 1.00, respectively).

F Modeling choices

The logistic functions $\alpha(t)$ and $\delta(t)$, used to model cell growth, take the value 1 at $t = 0$ because the normalized cell count is initialized at 1. Setting the asymptote parameter a_α or a_δ to a low value allows a drug with a higher growth rate k_α or k_δ to produce a faster but overall lower effect such that a drug with delayed (lower growth rate) but overall stronger effect (higher asymptotes) can surpass the initially faster, less effective drugs given sufficient time.

In the 4-parameter logistic curve, the maximum slope, given by the derivative at the inflection point $d = \gamma$, is proportional to $|\alpha - \delta|$ with proportionality factor $k_\beta = \frac{\beta \cdot \ln(10)}{4}$. This relationship motivates the choice of $\beta(t)$.

Regarding $\gamma(t)$, it is biologically plausible for the IC50 (or EC50) to change over time because of drug degradation and cellular adaptation mechanisms that commonly develop during extended exposure periods. However, modeling $\gamma(t)$ as a linearly or logistically decreasing function leads to overfitting on the data at hand. A possible reason for the strong performance with a constant $\gamma(t)$ is that at early time points, drug effects are minimal or have not yet manifested across all tested concentrations, meaning that cells initially grow at similar rates regardless of concentration. This results in a flat dose-time-response surface at early time points, i.e., $\alpha(t) - \delta(t) \approx 0$. Consequently, $f(d, t) \approx \delta(t)$, making the model effectively independent of $\gamma(t)$ at early time points. Thus, although $\gamma(t)$ is theoretically expected to be higher at early time points, violating this assumption has little practical impact.

Chapter 4

Discussion and conclusions

This chapter revisits the main findings of the three primary publications, considers aspects that have not yet been examined, and outlines directions for future work.

4.1 Summary

The first main contribution of this cumulative dissertation is a review of existing multi-output SVR models. It demonstrated that none of them were able to outperform single-output SVR models. Beyond this finding, the study also discussed the unsuitability of SVR methods for multi-output regression: Incorporating the vast number of constraints required to capture both linear and non-linear correlations among all output pairs into the quadratic program of an SVR model would quickly become computationally infeasible.

Next, the second main contribution of this cumulative dissertation critically assessed prevailing practices in drug response prediction, identifying significant limitations in modeling approaches as well as input and output datasets. Notably, TGSA, which at the time was considered the best-performing model, failed to surpass the two simple baselines, MMLP and the mean predictor, indicating that TGSA extracted no meaningful information from the chemical data. In addition, inconsistencies were observed within and across biological datasets as well as in current drug response metric data. Consequently, the study suggests that future models should consider fundamentally different data, incorporating the three-dimensional (3D) structures of both drug and protein, since most drugs exert their effects through binding to target proteins.

As a first step toward addressing these issues, particularly the shortcomings of current drug response metrics, the third main contribution of this cumulative dissertation introduced VUScope, a novel dose-time-response model designed for live-cell imaging data, along with GRIVUS, a corresponding metric that captures drug effects not only across concentration, but additionally across time. Although not reported in the third publication, it is worth mentioning that no prior study has been identified which, like VUScope, accurately reflects the absence of a response at the start of an experiment regardless of concentration; the only other dose-time-response model found, namely the model by Dhruba et al. [56], does not capture this feature (see their Figure 1). Additionally, since the time-dependent GR metric from Hafner et al. [22] is not truly time-dependent (see Section 2.4.4), there is no known precedent for a drug response metric that accounts for both dose and time dependence, making GRIVUS the first dose-time-dependent metric.

4.2 Discussion

The VUScope model builds upon the 4PL curve, extending it by replacing the four constant parameters with time-dependent functions:

$$f(d,t) = \frac{\alpha(t) - \delta(t)}{1 + 10^{\beta(t)\cdot(d-\gamma(t))}} + \delta(t).$$

This formulation can be adapted to different datasets (e.g., with dead cells excluded from the count) by adjusting the time-dependent functions. In its current implementation for the data at hand, $\gamma(t)$ is treated as a constant, while $\alpha(t)$ and $\delta(t)$ are modeled using a logistic function of the form $\frac{a\cdot 2^{k\cdot t}}{(a-1)+2^{k\cdot t}}$, where a is the upper asymptote and k is the growth rate, with the function starting at an initial value of 1 when $t = 0$. The choice for $\beta(t)$ is determined by $\alpha(t)$ and $\delta(t)$, since in a 4PL curve, the steepness parameter β is directly proportional to $|\alpha - \delta|$, as follows from the derivative of the 4PL curve at the inflection point $d = \gamma$, where the slope is steepest. Regarding $\gamma(t)$, although $\gamma(t)$ is theoretically expected to decrease over time, violating this assumption has little practical impact, which is discussed in the third primary publication (Section 3.3.2). Therefore, $\beta(t)$ and $\gamma(t)$ are not discussed further.

Two alternatives for $\alpha(t)$ and $\delta(t)$ were considered and evaluated. Assuming neither nutrient depletion nor spatial constraints, a simple exponential function $2^{k\cdot t}$ was considered for $\alpha(t)$ and $\delta(t)$. Although this assumption precludes the modeling of delayed drug efficacy (see Figure 3.2), it allows for attributing any drug response to the cell line-drug interaction only, excluding external factors. The Gompertz function [57] was also considered for $\alpha(t)$ and/or $\delta(t)$; it is commonly used to model population dynamics such as tumor growth [58] and is a sigmoidal function of the form $a^{1-2^{-k\cdot t}}$, where a is the upper asymptote and k is the growth rate, with the function starting at an initial value of 1 when $t = 0$. However, both alternatives resulted in poorer extrapolation over time for the available data, since GRIVUS normalization depends strongly on $\alpha(t)$. The exponential function grows too rapidly, causing $\alpha(t)$ to overpredict at $t = 120$, while the Gompertz function plateaus too early, causing $\alpha(t)$ to change too little from $t = 72$ to $t = 120$. Furthermore, the parameters of the Gompertz function are less biologically interpretable. In contrast to the base 2 and exponent $k \cdot t$ of the logistic function, which allow k to be interpreted as the initial doubling rate during the early, approximately exponential phase, the base a and exponent $1 - 2^{-k\cdot t}$ of the Gompertz function are less straightforward to interpret in a biological context.

All in all, the currently employed functions for VUScope are well-suited for the data at hand. To enable a more robust evaluation of VUScope, the dataset should be expanded. The most efficient approach would be to incorporate data from kinetic assays for HTS, which are fully compatible since VUScope requires only cell counts per well rather than the single-cell information provided by live-cell imaging. However, evaluating immunotherapy, which involves both immune and cancer cells, requires live-cell imaging, since standard kinetic assays for HTS capture only aggregated measurements per well and therefore cannot distinguish between the two cell types. Note that the current data in the third primary publication (Section 3.3.2) cover small-molecule compounds but do not include the latest immunotherapies.

4.3 Conclusions

Future work in drug response prediction should begin with the collection of spatial 3D data for both proteins and drugs. This data can then be used as input for existing or newly developed models that predict protein-ligand binding. For each cancer cell line, only binding events involving proteins that are actually present need to be considered. To determine protein presence, proteomics data should be binarized, with missing proteomics values imputed using EXP values as a proxy. These results can be extended to drug response prediction by linking potential binding events to specific outcomes, such as low or high GRIVUS (or a traditional metric like AUC where GRIVUS is not available), while also incorporating protein abundance through the actual proteomics (or EXP) values rather than binarized ones. For non-ligand drugs, using 3D membrane structures instead of 3D protein structures may capture their pharmacodynamics.

In conclusion, although considerable challenges remain, the integration of the approaches holds significant potential to advance the field of drug response prediction, taking an important step toward the original goal: not merely increasing the number of scientific publications, but genuinely making a meaningful positive impact in patients' lives.

References

[1] R Arafeh, T Shibue, JM Dempster, et al. “The present and future of the Cancer Dependency Map”. In: *Nature Reviews Cancer* 25.1 (2025), pp. 59–73.

[2] M Ghandi, FW Huang, J Jané-Valbuena, et al. “Next-generation characterization of the cancer cell line encyclopedia”. In: *Nature* 569.7757 (2019), pp. 503–508.

[3] W Yang, J Soares, P Greninger, et al. “Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells”. In: *Nucleic Acids Research* 41.D1 (Nov. 2012), pp. D955–D961. ISSN: 0305-1048.

[4] D van der Meer, S Barthorpe, W Yang, et al. “Cell Model Passports—a hub for clinical, genetic and functional datasets of preclinical cancer models”. In: *Nucleic Acids Research* 47.D1 (2019), pp. D923–D929.

[5] S Kim, J Chen, T Cheng, et al. “PubChem 2025 update”. In: *Nucleic Acids Research* 53.D1 (Nov. 2024), pp. D1516–D1525. ISSN: 1362-4962.

[6] BJ Bryant and KM Knights. *Pharmacology for health professionals*. Elsevier Australia, 2011.

[7] M Weatherall. “The meaning and importance of drug potency in medicine”. In: *Clinical Pharmacology & Therapeutics* 7.5 (1966), pp. 577–582.

[8] NHG Holford and LB Sheiner. “Understanding the Dose-Effect Relationship”. In: *Clinical Pharmacokinetics* 6.6 (1981), pp. 429–453. ISSN: 1179-1926.

[9] Technology Networks. *High Throughput Screening – Accelerating Drug Discovery Efforts*. 2021. URL: https://technologynetworks.com/tn/articles/high-throughput-screening-accelerating-drug-discovery-efforts-288926 (visited on 08/22/2025).

[10] Sartorius. *Incucyte® S3 Live-Cell Analysis Instrument*. 2025. URL: https://sartorius.com/en/products/live-cell-imaging-analysis/live-cell-analysis-instruments/s3-live-cell-analysis-instrument (visited on 08/22/2025).

[11] Promega Corporation. *CellTiter-Glo Luminescent Cell Viability Assay*. Technical Bulletin 288. Promega Corporation. 2023. URL: https://www.promega.de/-/media/files/resources/protocols/technical-bulletins/0/celltiter-glo-luminescent-cell-viability-assay-protocol.pdf (visited on 06/10/2025).

[12] Promega Corporation. *GloMax Discover System*. Technical Manual 397. Promega Corporation. 2024. URL: https://www.promega.de/-/media/files/resources/protocols/technical-manuals/101/glomax-discover-system-protocol.pdf (visited on 06/10/2025).

[13] DJ Vis, L Bombardelli, H Lightfoot, et al. “Multilevel models improve precision and speed of IC_{50} estimates”. In: *Pharmacogenomics* 17.7 (2016), pp. 691–700.

[14] Promega Corporation. *CellTiter-Glo® 2.0 Cell Viability Assay*. 2025. URL: https://sartorius.com/en/products/live-cell-imaging-analysis/live-cell-analysis-instruments/s3-live-cell-analysis-instrument (visited on 08/22/2025).

[15] PG Gottschalk and JR Dunn. “The five-parameter logistic: a characterization and comparison with the four-parameter logistic”. In: *Analytical Biochemistry* 343.1 (2005), pp. 54–65.

[16] J Macdougall. “Analysis of Dose–Response Studies—E_{max} model”. In: *Dose Finding in Drug Development*. Springer, 2006, pp. 127–145.

[17] MA Branch, TF Coleman, and Y Li. “A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems”. In: *SIAM Journal on Scientific Computing* 21.1 (1999), pp. 1–23.

[18] P Virtanen, R Gommers, TE Oliphant, et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272.

[19] AV Hill. “The possible effects of the aggregation of the molecules of hemoglobin on its dissociation curves”. In: *The Journal of Physiology* 40 (1910), pp. iv–vii.

[20] J Barretina, G Caponigro, N Stransky, et al. “The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity”. In: *Nature* 483.7391 (2012), pp. 603–607.

[21] B Haibe-Kains, N El-Hachem, NJ Birkbak, et al. “Inconsistency in large pharmacogenomic studies”. In: *Nature* 504.7480 (2013), pp. 389–393.

[22] M Hafner, M Niepel, M Chung, et al. “Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs”. In: *Nature Methods* 13.6 (2016), pp. 521–527.

[23] M Fallahi-Sichani, S Honarnejad, LM Heiser, et al. “Metrics other than potency reveal systematic variation in responses to cancer drugs”. In: *Nature Chemical Biology* 9.11 (2013), pp. 708–714.

[24] Broad Institute. *DepMap Portal*. 2025. URL: https://depmap.org/portal/ (visited on 07/11/2025).

[25] Cancer Genome Project. *CaVEMan*. Genome Research Ltd., 2015. URL: https://cancerit.github.io/CaVEMan/ (visited on 07/14/2025).

[26] K Ye, MH Schulz, Q Long, et al. “Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads”. In: *Bioinformatics* 25.21 (2009), pp. 2865–2871.

[27] D Benjamin, T Sato, K Cibulskis, et al. “Calling somatic SNVs and indels with Mutect2”. In: *bioRxiv* (2019), p. 861054.

[28] GA Van der Auwera, MO Carneiro, C Hartl, et al. “From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline”. In: *Current Protocols in Bioinformatics* 43.1 (2013), pp. 11–10.

[29] CD Greenman, G Bignell, A Butler, et al. “PICNIC: an algorithm to predict absolute allelic copy number variation with microarray cancer data”. In: *Biostatistics* 11.1 (2010), pp. 164–175.

[30] CH Mermel, SE Schumacher, B Hill, et al. “GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers”. In: *Genome Biology* 12.4 (2011), R41.

[31] M Riester, AP Singh, AR Brannon, et al. “PureCN: copy number calling and SNV classification using targeted short read sequencing”. In: *Source Code for Biology and Medicine* 11.1 (2016), p. 13.

[32] Diakumis P. *PURPLE*. Hartwig Medical Foundation, 2024. URL: https://umccr.github.io/gpgr/articles/purple.html (visited on 07/14/2025).

[33] SL Carter, K Cibulskis, E Helman, et al. “Absolute quantification of somatic DNA alterations in human cancer”. In: *Nature Biotechnology* 30.5 (2012), pp. 413–421.

[34] GATK Team. *Panel of Normals (PON)*. Broad Institute, 2024. URL: https://gatk.broadinstitute.org/hc/en-us/articles/360035890631-Panel-of-Normals-PON (visited on 07/15/2025).

[35] Illumina. *Infinium® HumanMethylation450 BeadChip*. 2025. URL: https://www.illumina.com/content/dam/illumina-marketing/documents/products/datasheets/datasheet_humanmethylation450.pdf (visited on 07/11/2025).

[36] MD Robinson and A Oshlack. “A scaling normalization method for differential expression analysis of RNA-seq data”. In: *Genome Biology* 11.3 (2010), R25.

[37] M Love, S Anders, W Huber, et al. “Differential analysis of count data–the DESeq2 package”. In: *Genome Biology* 15.550 (2014), pp. 10–1186.

[38] Y Zhao, MC Li, MM Konaté, et al. “TPM, FPKM, or normalized counts? A comparative study of quantification measures for the analysis of RNA-seq data from the NCI patient-derived models repository”. In: *Journal of Translational Medicine* 19.1 (2021), p. 269.

[39] B Li and CN Dewey. “RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome”. In: *BMC Bioinformatics* 12.1 (2011), p. 323.

[40] R Patro, G Duggal, MI Love, et al. “Salmon provides fast and bias-aware quantification of transcript expression”. In: *Nature Methods* 14.4 (2017), pp. 417–419.

[41] J Cox, MY Hein, CA Luber, et al. “Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ”. In: *Molecular & Cellular Proteomics* 13.9 (2014), pp. 2513–2526.

[42] DP Nusinow, J Szpyt, M Ghandi, et al. “Quantitative proteomics of the cancer cell line encyclopedia”. In: *Cell* 180.2 (2020), pp. 387–402.

[43] D Szklarczyk, R Kirsch, M Koutrouli, et al. “The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest”. In: *Nucleic Acids Research* 51.D1 (2023), pp. D638–D646.

[44] P Shannon, A Markiel, O Ozier, et al. “Cytoscape: a software environment for integrated models of biomolecular interaction networks”. In: *Genome research* 13.11 (2003), pp. 2498–2504.

[45] D Szklarczyk, A Santos, C Von Mering, et al. “STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data”. In: *Nucleic Acids Research* 44.D1 (2016), pp. D380–D384.

[46] X Liu, D Feng, J Chen, et al. “HCDT 2.0: A Highly Confident Drug-Target Database for Experimentally Validated Genes, RNAs, and Pathways”. In: *Scientific Data* 12.1 (2025), p. 695.

[47] B Zdrazil, E Felix, F Hunter, et al. “The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods”. In: *Nucleic Acids Research* 52.D1 (Nov. 2023), pp. D1180–D1192. ISSN: 0305-1048.

[48] P Walters. *PubChemPy: A Python Wrapper for the PubChem PUG REST API*. 2014. URL: https://github.com/mcs07/PubChemPy.

[49] G Landrum, P Tosco, B Kelley, et al. *rdkit/rdkit: 2025_03_4 (Q1 2025) Release (Release_2025_03_4)*. Version Release_2025_03_4. 2025. URL: https://doi.org/10.5281/zenodo.591637.

[50] B Ramsundar, P Eastman, P Walters, et al. *Deep Learning for the Life Sciences*. O’Reilly Media, 2019.

[51] Y Li, DE Hostallero, and A Emad. “Interpretable deep learning architectures for improving drug response prediction performance: myth or reality?” In: *Bioinformatics* 39.6 (2023), btad390.

[52] D Diaz-Vico, J Prada, A Omari, et al. “Deep support vector neural networks”. In: *Integrated Computer-Aided Engineering* 27.4 (2020), pp. 389–402.

[53] RI Tange, MA Rasmussen, E Taira, et al. “Benchmarking support vector regression against partial least squares regression and artificial neural network: Effect of sample size on model performance”. In: *Journal of Near Infrared Spectroscopy* 25.6 (2017), pp. 381–390.

[54] B Shen, F Feng, K Li, et al. “A systematic assessment of deep learning methods for drug response prediction: from in vitro to clinical applications”. In: *Briefings in Bioinformatics* 24.1 (2023), bbac605.

[55] Y Zhu, Z Ouyang, W Chen, et al. “TGSA: protein–protein association-based twin graph neural networks for drug response prediction with similarity augmentation”. In: *Bioinformatics* 38.2 (2022), pp. 461–468.

[56] SR Dhruba, A Rahman, R Rahman, et al. “Recursive model for dose-time responses in pharmacological studies”. In: *BMC Bioinformatics* 20 (2019), pp. 1–12.

[57] B Gompertz. “XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. F. R. S. &c”. In: *Philosophical Transactions of the Royal Society of London* 115 (1825), p. 518.

[58] C Vaghi, A Rodallec, R Fanciullino, et al. “Population modeling of tumor growth curves and the reduced Gompertz model improve prediction of the age of experimental tumors”. In: *PLoS Computational Biology* 16.2 (2020), e1007178.