Technische Universität München
TUM School of Natural Sciences

Improving Gradient Boosting Machine for Modelling High Throughput Screening Data in Drug Discovery

Davide Boldini

Vollständiger Abdruck der von der TUM School of Natural Sciences der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Cathleen Zeymer

Prüfende der Dissertation:

1. Prof. Dr. Stephan A. Sieber
2. Prof. Dr. Angela Casini
3. Prof. Dr. Andrea Volkamer

Die Dissertation wurde am 09.02.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Natural Sciences am 10.04.2024 angenommen.

"Wherever there is building up, love is present, and wherever love is, there is building up"

Works of Love, Søren Kierkegaard

Abstract

Despite the social and economic impact of pharmacological research, the drug discovery pipeline remains an extremely lengthy, complex and expensive process. Recent estimates show that on average, the research and development cycle required to produce a new drug costs $1.3 billion over a 9 year period. Additionally, only 10% of all drug discovery campaigns succeeds in delivering a new therapeutic to the market, thus making pharmacological research an economic high-risk investment.

In this context, machine learning (ML) can exploit the large amounts of data collected during previous High Throughput Screening (HTS) campaigns to provide accurate *in silico* bioactivity predictions, thus accelerating the discovery of new therapeutics. While there is a wealth of different ML algorithms for Quantitative Structure-Activity Relationship (QSAR) modelling, constructing reliable and efficient ML-based predictors from HTS data continues to be a challenging endeavor. This is because these datasets are mostly comprised of inactive molecules, have a large number of false positives, are extremely large and are often not sufficiently chemically diverse.

In this thesis, a broad range of computational methods are developed and investigated to tackle these four issues, focusing specifically on the Gradient Boosting Machine (GBM) algorithm for modelling HTS datasets. A new approach is presented in Chapter 3 to tackle the issue of class imbalance by adjusting the learning objective of GBM to account for the dataset bias towards inactive compounds. These modifications push GBM to match or outperform state-of-the-art QSAR predictors on a variety of benchmarks, while increasing its computational efficiency. In Chapter 4, the three main variants of the GBM algorithm are benchmarked for molecular property prediction. The analysis shows that different GBM implementations have different strengths on large datasets, e.g. XGBoost has superior performance while LightGBM is orders of magnitude faster than the others. Additionally, the set of most important parameters to optimize is established for the GBM algorithm. In Chapter 5, a new data valuation algorithm for GBM is developed and its application for the efficient prioritization of HTS hits is investigated. The proposed approach demonstrates promising performance in false positive and true positive identification on a broad range of HTS datasets and could be successfully employed on a retrospective case study. Finally, Chapter 6 investigates the use of molecular fingerprints for modelling natural products in terms of virtual screening and bioactivity prediction. As such, the results of this benchmark provide featurization guidelines to consider when training QSAR models on HTS data, if the goal is to perform virtual screening beyond the drug-like chemical space.

In conclusion, this thesis provides novel insights and algorithms for training QSAR models on HTS data using GBMs. These findings provide ready-to-use solutions for accelerating early-stage drug discovery and act as a foundation for further research in the modelling of this class of datasets.

Kurzfassung

Die Arzneimittelforschung ist trotz der sozialen und wirtschaftlichen Bedeutung nach wie vor ein äußerst langwieriger, komplexer und teurer Prozess. Jüngsten Schätzungen zufolge kostet der Forschungs- und Entwicklungszyklus, der zur Herstellung eines neuen Medikaments erforderlich ist, über einen Zeitraum von neun Jahren durchschnittlich 1,3 Milliarden Dollar. Darüber hinaus gelingt es nur in 10 % der Fälle, ein neues Therapeutikum auf den Markt zu bringen. Das macht die pharmakologische Forschung zu einer ökonomisch riskanten Investition.

In diesem Zusammenhang kann Machine Learning (ML) die großen Datenmengen nutzen, die bei früheren Hochdurchsatz-Screening-Kampagnen (engl. HTS) gesammelt wurden, um genaue *in silico* Bioaktivitätsvorhersagen zu erstellen und so die Entdeckung neuer Therapeutika zu beschleunigen. Es gibt viele verschiedene ML-Algorithmen zur Modellierung quantitativer Struktur-Wirkungs-Beziehungen (engl. QSAR), aber die Entwicklung zuverlässiger und effizienter ML Modellen aus HTS-Daten ist nach wie vor ein komplexes Unterfangen. Dies liegt daran, dass diese Datensätze größtenteils aus inaktiven Molekülen bestehen, eine große Anzahl von falsch-positiven Ergebnissen aufweisen, außerordentlich groß und oft chemisch nicht ausreichend vielfältig sind.

In dieser Arbeit wird ein breites Spektrum an Methoden entwickelt und untersucht, um diese vier Probleme anzugehen, wobei der Schwerpunkt auf dem Gradient Boosting Machine (GBM)-Algorithmus zur Modellierung von HTS-Datensätzen liegt. In Kapitel 3 wird ein neuer Ansatz vorgestellt, um das Problem des Klassenungleichgewichts anzugehen, indem das Lernziel der GBM so angepasst wird, dass die Verzerrung des Datensatzes hin zu inaktiven Verbindungen berücksichtigt wird. Diese Modifikationen ermöglichen es mit GBM, bei einer Reihe von Benchmarks mit den modernsten QSAR-Modellen zu konkurrieren oder diese sogar zu übertreffen, während gleichzeitig die Berechnungseffizienz erhöht wird. Kapitel 4 unterzieht drei Varianten des GBM-Algorithmus einer Benchmark-Analyse zur Vorhersage von Moleküleigenschaften. Die Analyse zeigte, dass die verschiedenen GBM-Implementierungen bei großen Datensätzen unterschiedliche Stärken aufweisen, z. B. hat XGBoost eine überlegene Leistung, während LightGBM deutlich schneller ist als die anderen. Außerdem wurden die wichtigsten zu optimierenden Parameter für den GBM-Algorithmus ermittelt. Kapitel 5 beschäftigt sich mit der Entwicklung eines neuen Datenbewertungsalgorithmus für GBM und dessen Anwendung zur effizienten Priorisierung von HTS-Treffern. Der vorgeschlagene Ansatz zeigte eine vielversprechende Leistung bei einem breiten Spektrum von HTS-Datensätzen und konnte erfolgreich bei einer retrospektiven Fallstudie eingesetzt werden. Schließlich wird in Kapitel 6 die Verwendung von molekularen Fingerabdrücken für die Modellierung von Naturstoffen im Hinblick auf virtuelles Screening und Bioaktivitätsvorhersage untersucht. Die Ergebnisse liefern Richtlinien für die computergestützte Darstellung von Molkülen, die beim Training von QSAR-Modellen auf HTS-Daten zu berücksichtigen

sind, wenn das Ziel darin besteht, ein virtuelles Screening außerhalb der arzneimittelähnlichen chemischen Verbindungen hinaus durchzuführen.

Zusammenfassend liefert diese Arbeit neue Erkenntnisse und Algorithmen für das Training von QSAR-Modellen auf HTS-Daten unter Verwendung von GBMs liefert. Diese Erkenntnisse bieten gebrauchsfertige Lösungen zur Beschleunigung der Arzneimittelentdeckung im Frühstadium und dienen als Grundlage für die weitere Forschung im Bereich der Modellierung dieser Klasse von Datensätze.

Abbreviations

ADMET	Absorption, Distribution, Metabolism, Excretion and Toxicity
HTS	High Throughput Screening
AI	Artificial Intelligence
GBM	Gradient Boosting Machine
NP	Natural Product
ML	Machine Learning
DL	Deep Learning
VS	Virtual Screening
GNN	Graph Neural Network
KNN	k-Nearest Neighbors
SVM	Support Vector Machine
RF	Random Forest
NN	Neural Network
MPNN	Message Passing Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
AFP	Attentive Fingerprints
LSTM	Long Short Term Memory
CNN	Convolutional Neural Network
QSAR	Quantitative Structure-Activity Modelling
CV	Computer Vision
NLP	Natural Language Processing
TP	True Positive
FP	False Positive
TN	True Negative

FN	False Negative
ROC-AUC	Receiver Operator Characteristic Area Under Curve
PR-AUC	Precision-Recall Area Under Curve
MCC	Matthews Correlation Coefficient
BEDROC	Boltzmann Enhanced Discrimination of ROC
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
WHIM	Weighted Holistic Invariant Molecular
WHALES	Weighted Holistic Atom Localization and Entity Shape
ACM	Atom Centered Mahalanobis
ASP	All-Shortest Paths
AP	Atom Pair
DFS	Depth-First Search
ECFP	Extended Connectivity Fingerprint
FCFP	Functional Class Fingerprint
PP2	Pharmacophore Pairs
MHFP	Minhashed Fingerprint
MAP4	Minhashed Atom Pair
LogS	Solubility
MW	Molecular Weight
LogP	Octanol-water partition coefficient
RB	Rotatable Bonds
EFB	Exclusive Feature Bundling
GOSS	Gradient-based One Sided Sampling
TS	Target Statistics
SMOTE	Synthetic Minority Oversampling Technique

ADASYN	Adaptive Synthetic Oversampling
GHOST	Generalized Threshold Shifting
IF	Influence Function
DS	Data Shapley
MVS-A	Minimum Variance Sampling Analysis
CHT	Presynaptic Choline Transporter
COCONUT	Collection of Open Natura Products
CMPND	Comprehensive Marine Natural Products Database

Parts of this thesis have been published or submitted in peer-reviewed journals as listed below:

Machine learning assisted hit prioritization for high throughput screening in drug discovery

Davide Boldini, Lukas Friedrich, Daniel Kuhn, Stephan A. Sieber

*ACS Cent. Sci. (**2024**)*

https://doi.org/10.1021/acscentsci.3c01517

Effectiveness of molecular fingerprints for exploring the chemical space of natural products

Davide Boldini, Davide Ballabio, Viviana Consonni, Roberto Todeschini, Francesca Grisoni, Stephan A. Sieber

*J Cheminform 16, 35 (**2024**)*

https://doi.org/10.1186/s13321-024-00830-3

Practical guidelines for the use of gradient boosting for molecular property prediction

Davide Boldini, Francesca Grisoni, Daniel Kuhn, Lukas Friedrich, Stephan A. Sieber

*J Cheminform 15, 73 (**2023**)*

https://doi.org/10.1186/s13321-023-00743-7

Tuning gradient boosting for imbalanced bioassay modelling with custom loss functions

Davide Boldini, Lukas Friedrich, Daniel Kuhn, Stephan A. Sieber

*J Cheminform 14, 80 (**2022**)*

https://doi.org/10.1186/s13321-022-00657-w

Parts of this thesis have been presented at conferences and workshops:

Colloquium Chemometricum Mediterraneum 2023

"*Gradient boosting for molecular property prediction*", Davide Boldini

27th – 30st June 2023, Padova, Italy

oral presentation

Further work performed during the doctorate but not included in the thesis:

Synergising Chemical Structures and Bioassay Descriptions for Enhanced Molecular Property Prediction in Drug Discovery

Maximilian Schuh**, Davide Boldini**, Stephan A. Sieber

Under revision in the Journal of Chemical Information and Modelling. Currently available as a preprint on arXiv.

arXiv:2401.04478

Data Valuation: A novel approach for analyzing high throughput screen data using machine learning

Joshua Hesse**, Davide Boldini**, Stephan A. Sieber

To be submitted to the Journal of Chemical Information and Modelling. Currently available as a preprint on ChemRxiv.

10.26434/chemrxiv-2023-wlzlc

Contents

Part I - Introduction

Part II - Publications

Part III - Appendix

PART I

INTRODUCTION

1.
Scientific background

1.1 A perspective on the societal and economic impact of drug discovery

Small-molecule pharmacological research has been one of the main contributing factors to the economic and societal improvement observed across the planet during the 20th and 21st century.[1] For example, the discovery of Penicillin in 1928 by Alexander Fleming, together with the development of Salvarsan (1910) and Prontosil (1935), kickstarted the "golden age of antibiotics", drastically reducing the number of deaths associated to bacterial infection.[1,2] Another example is the development of Retrovir (1987), the first effective treatment for tackling HIV infections.[1] After further research on small-molecule antiviral agents, what once was considered an epidemic is nowadays a manageable chronic condition with good quality of life.[3] In terms of its societal impact, pharmaceutical research has contributed to more than 60% of the life expectancy increase in the last 20 years,[4] while from an economic point of view it provides per year a total gross value of approximately $530 billion and employs over 5.5 million people.[5]

To ensure drug safety and efficacy, the development pipeline is usually divided into four phases (Figure 1.1):[6,7]

1. **Early drug discovery (phase 0):** Here, the goal is to identify a relevant biological target for the disease of interest and find a selection of compounds that elicit the desired response (e.g. protein inhibition). Next, the most active molecules undergo further optimization, usually with the aim of improving Absorption, Distribution, Metabolism, Excretion and Toxicity (ADMET) properties. Finally, the first *in vivo* tests are performed on animal models.[8]
2. **Phase 1:** The most promising compound is then evaluated on a small cohort of healthy volunteers for further pharmacokinetic optimization and establishing dosage ranges.[9,10]
3. **Phase 2:** Initial testing is carried out on a small selection of patients in order to establish therapeutic dosage and estimate the efficacy of the drug against the target disease.[11]
4. **Phase 3:** Additional drug efficacy testing is carried out involving a much larger cohort of patients and centers. All instances of adverse effects are recorded and evaluated.[12]

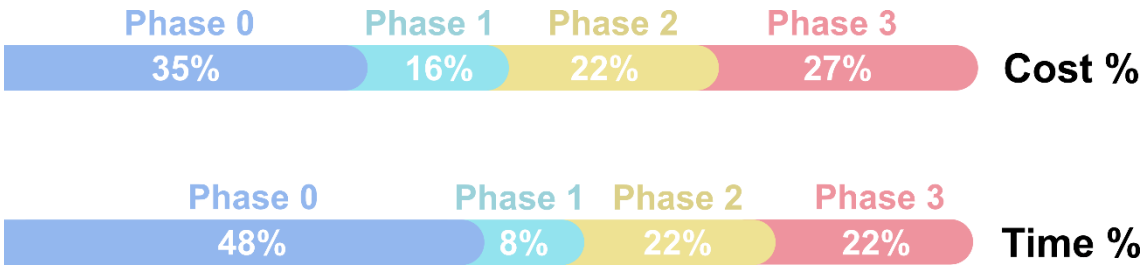


Figure 1.1 – Analysis of the contribution each development phase has on the overall budget and development time required to deliver a drug on the market. Data adapted from Ref. 6.

Once a drug successfully undergoes Phase 3, the pharmaceutical company can request regulatory approval and begin selling it on the market.[6,8] Due to its complexity however, the drug discovery pipeline is a particularly lengthy, costly and unpredictable endeavor.[6,13] Recent estimates indicate that for each new therapeutic to reach the market, companies need to invest on average $1.3 billion over a 9 year period.[13–15] Additionally, only 10% of all drug discovery campaigns succeed in producing a marketable product, further exacerbating the time and monetary investment required for research and development.[6,16,17] While researchers have investigated the impact of the regulatory and legal framework surrounding the development pipeline,[18,19] the discovery of new therapeutics remains the main bottleneck in pharmaceutical research.[20,21] Because of these reasons, there is a significant interest in leveraging new technologies that can help efficiently identify the most promising compounds for drug discovery campaigns.

In this context, two key innovations have disrupted the drug discovery pipeline. On the experimental side, high throughput screening (HTS), initially introduced in the early 2000s, has revolutionized the way early stage drug discovery is conducted by enabling fast screening of hundreds of thousands of compounds for biological activity.[22,23] On the computational side, artificial intelligence (AI) has recently started gaining traction in the pharmaceutical field, enabling *in silico* molecular property prediction, target identification, *de novo* drug design and automated synthesis planning.[24,25]

While these two techniques naturally complement each other, given that one allows to generate large amounts of data efficiently while the other improves as the dataset size increases, their combination is not straightforward. The aim of this thesis then is to investigate how to optimally model HTS data with Gradient Boosting Machine (GBM), a promising machine-learning algorithm for molecular property prediction. Developing accurate predictive models from the large amount of historical HTS data available today is essential for expediting the identification of relevant bioactive molecules, thus reducing the time and costs associated with drug discovery campaigns.

1.2 High throughput screening in drug discovery

A HTS campaign consists of measuring a given biochemical property, e.g. a desired phenotypic response or target protein inhibition, for a large number of compounds by miniaturizing the assay and leveraging robotic equipment.[26] Experiments are usually performed without replicates at a single concentration (e.g. 10 µM), using less than 2 µL per well.[27,28] The use of robotic equipment, single measurements and low volumes enables fast and cheap screening of compound libraries up to 10^7 molecules.[29] However, only a small fraction of screened compounds show meaningful bioactivity and many hits are false positives, thus limiting the effectiveness of HTS campaigns for identifying new drug candidates.[22,30]

The low hit rate of HTS campaigns (<1% on average) is typically a consequence of the choice of compounds to screen.[29] As such, library design for HTS is an active topic of research.[31,32] In the early 2000s, the majority of screening libraries prominently featured chemical moieties which were easily accessible via combinatorial chemistry. However, doing so heavily limited the success rate of screening campaigns, given the lack of novelty and diversity of combinatorial compound libraries.[29] Since then, a premium has been placed on overall molecular diversity, with the aim of covering as much chemical space as possible, and on focusing on compounds that have *a priori* high chance of being active against the target of interest.[33] Recently, efforts have been made to include natural products (NPs) in screening libraries, due to their potency, selectivity and chemical diversity.[34]

Concerning HTS false positives, one common noise source is related to specific well positions (e.g. under or overestimation of the readout for wells at the edge of the plate, also known as the “edge effect”), or to particular plates (e.g. incubation time drift).[26,28] Typically, these sources of error are tackled by well position and plate order randomization in subsequent screens, readout normalization according to positive and negative controls for the assay, position normalization via correction methods and outlier detection via statistical testing. Each of these approaches has advantages and disadvantages, making it necessary to analyse each HTS on a case-by-case basis.[26,28]

Another prevalent source of noise in HTS campaigns are compounds that elicit a reproducible readout which does not correlate with the underlying biological activity the assay aims to measure. Examples of this class are autofluorescent compounds, colloidal aggregators and assay technology interferents (e.g. Firefly Luciferase binders).[30,35,36] Additionally, there are compounds that tend to be active in most HTS campaigns regardless of biological target or assay technology. These molecules, named “frequent

hitters", are sometimes false positives, in the sense that they present a fluorescence readout that is not correlated with the desired biological response, while in other cases they are simply pharmacologically promiscuous.[37] Due to the heterogeneity of interference mechanisms and the interplay between chemical structure, biological target, assay technology and conditions, it is extremely difficult to establish only from primary HTS data which compound might be a false positive. As such, most drug discovery campaigns employ confirmatory screens (e.g. dose-response measurements) and counterscreens (e.g. repeating the assay in a different cell line) to identify the true hits among the active compounds in the primary HTS campaign.[38]

Stemming from these considerations, HTS datasets pose the following challenges for AI-based modelling:

- The low hit rate makes it difficult for data-driven algorithms to learn how to distinguish between active and inactive compounds. This issue is tackled in Chapter 3 of this thesis.
- The size of HTS datasets makes training and optimization of AI models challenging, making the identification of computationally efficient approaches a priority. This aspect is discussed in Chapter 4 of the dissertation.
- The prevalence of false positives in HTS campaigns reduces their effectiveness for data-driven modelling, since the algorithms will learn spurious correlations from the training data. A new method for distinguishing between true and false positives in HTS data is presented in Chapter 5 of this thesis.
- The chemical space explored by HTS campaigns might not be sufficiently diverse, thus limiting the applicability of data-driven models to other promising compound classes such as natural products. How to model natural products for molecular property prediction is discussed in Chapter 6.

1.3 Leveraging artificial intelligence to expedite drug discovery

Drug development is fundamentally a multi-objective optimization task, where the goal is to design a compound satisfying multiple constraints, e.g. sufficient potency and low toxicity.[25] Given the complexity of biological systems, the vastness of the chemical space and the high cost of experiments, *in silico* approaches are an ideal tool to support medicinal chemists in designing new drugs.[25,39] In the last decade, AI has risen to prominence in this regard, thanks to a rapid increase in computational power and the amount of data available for training data-driven methods.[24,40]

A cornerstone application of AI to drug discovery is molecular property prediction, defined as the task of predicting *in silico* molecular properties from the compound structure.[24] To create molecular property prediction models, it is necessary to first have a dataset of molecules (usually between 10^3 – 10^6) that have been measured in the assay of interest.[41] A machine learning (ML) or deep learning (DL) algorithm can then be used to learn the relationship between chemical structure and the desired property from the training data.[41,42] To obtain a broadly applicable predictive model, it is paramount that the training set includes a structurally diverse selection of compounds.[24,42]

1.3.1 Uses of molecular property prediction in drug discovery

Typical applications of molecular property prediction include ligand-based virtual screening (VS), active learning and ADMET property modelling.[41]

Ligand-based VS is typically employed to identify in large chemical libraries (up to 10^9 molecules) the most potent compounds for a given bioassay.[43,44] As such, it is used in the earliest stages of the drug discovery campaign in order to prioritize which compounds to develop further.[45] One crucial difference from structure-based VS methods such as docking is that they only rely on chemical information to predict bioactivity.[24] Therefore, as long as a dataset of sufficient quality has been acquired for the assay of interest, having a high-quality crystal structure of the biological target or knowledge of the binding mode of the ligands is not required. Because of this, molecular property prediction models can also be employed to predict phenotypic activity where the target might be unknown, e.g. antibacterial or antiviral activity.[46,47] One particularly notable success story of the use of ligand-based VS is the discovery of Halicin, a novel broad spectrum antibiotic.[46] Halicin was identified as a potential antibiotic by a Graph Neural Network (GNN)[48] while *in silico* screening of a drug repurposing library. The GNN was trained on a HTS dataset for *Escherichia coli* antibacterial activity which focused on chemical diversity and included several natural products to generate the best possible training set for the data-driven predictor. Crucially, the study showed that the GNN could identify

antibiotic activity beyond the scope of the original assay, as shown by the excellent performance of Halicin against other strains (e.g. *Mycobacterium tubercolosis*).

Active learning is conceptually similar to ligand-based VS, but differs in its role within a drug discovery campaign. While the latter is employed after a dataset has been acquired, the former is used to decide which compound to measure next as data is collected.[49–52] In practice, the active learning procedure goes as follows (Figure 1.2):

1. Measure a small number of compounds in the assay of interest.
2. Train a molecular property prediction model on the dataset acquired during step 1.
3. Predict which compounds are most likely to be active among the remaining molecules from the screening library.
4. Validate experimentally the top-*k* most promising compounds identified by the molecular property prediction model.
5. Train a new molecular property prediction model including the newly acquired data.
6. Repeat steps 3-5 until a sufficient number of bioactive compounds has been found.

This procedure significantly improves the hit rate of screening procedures, enabling detection of active compounds at a much faster rate than by random selection. Additionally, the application of active learning is not limited to *in vitro* screening campaigns, but can also be used to accelerate other more computationally demanding *in silico* approaches, such as docking.[50,53]

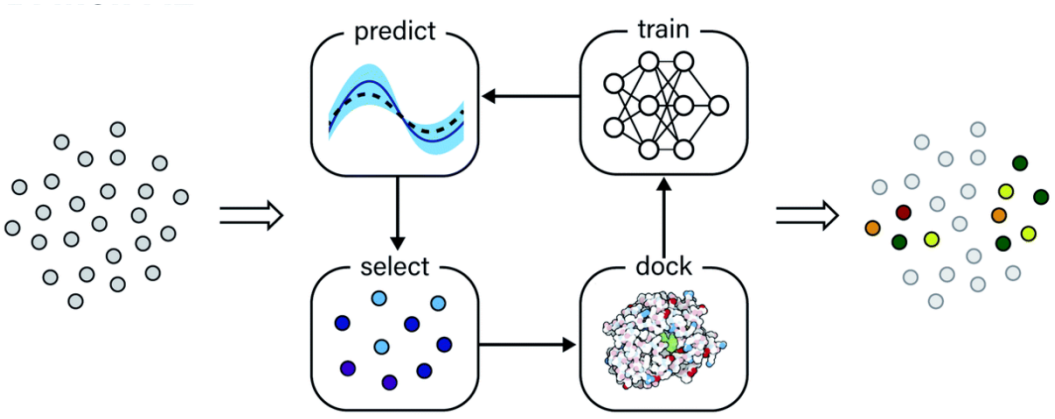


Figure 1.2 –Active learning workflow to accelerate structure-based virtual screening. Once a few molecules at random have been docked, the ML model is trained to predict docking scores. The predictor is then used to identify from the remaining screening library the most promising compounds, which are then validated via docking. By repeating this procedure iteratively, it is possible to identify the most promising structure-based virtual screening hits without needing to dock the entire library. Figure is adapted from Ref. 53.

Finally, ADMET prediction models are used to further refine the selection of bioactive compounds or to guide the pharmacological optimization of the most promising lead molecules.[54] As of late, these models are starting to be paired with DL-based generative

models.[55,56] By linking the two approaches together, it is possible to steer the generative process so that it focuses on pharmacologically desirable regions of the chemical space, thus producing more lead-like molecular libraries.[56]

1.3.2 Algorithms for molecular property prediction

Many different computational approaches are used for molecular property prediction (Figure 1.3). One of the first algorithms used for this task was *k*-Nearest Neighbors (KNN), where a compound's prediction is calculated on the basis of which compounds among the training set it is most similar.[57] As such, this algorithm relies on the similarity principle, stating that similar compounds tend to have similar molecular properties. Afterwards, the field moved from linear models (e.g. linear or logistic regression) to non-linear models such as Support Vector Machines (SVMs)[58] and decision tree ensembles such as Random Forest (RF)[59] in order to capture more complex dependencies between compound structures and molecular properties.[24] In the last decade, neural networks (NNs) rose to prominence, thanks to their ability to model multiple properties at once and model molecular structures without expert-encoded features.[41] However, there is no consensus yet on whether classical ML approaches such as SVM and RF are outperformed by neural networks.[60–62]

Finally, Gradient Boosting, a decision tree ensemble algorithm, has recently garnered the attention of the cheminformatics community, driven by its outstanding performance on tabular data modelling in different fields.[63–66] In terms of performance, it typically matches or outperforms both machine learning and deep learning algorithms while remaining computationally lightweight.[65,66] Additionally, its predictions are straightforward to explain in terms of feature contributions, e.g. via efficient Shapley value calculation.[67] Stemming from these considerations, the aim of this thesis is then to investigate how to adapt Gradient Boosting for modelling HTS data for molecular property prediction, so that it can effectively tackle the class imbalance, computational cost, false positive rate and chemical bias typical of this class of datasets.

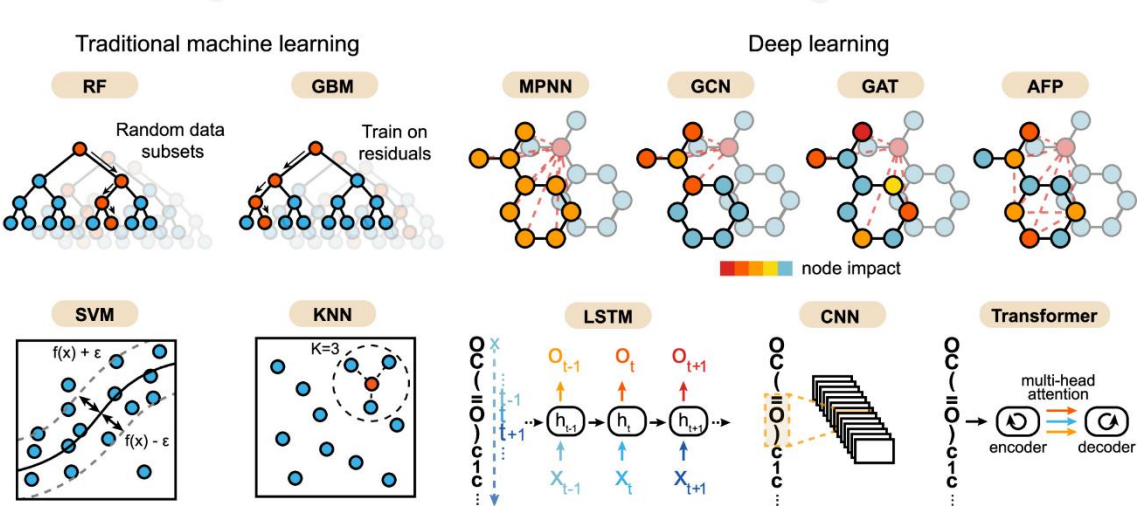


Figure 1.3 – Different machine learning and deep learning strategies commonly used for molecular property prediction. Among traditional machine learning algorithms, Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Machine (SVM) and *k*-Nearest Neighbours (KNN) are the most popular approaches. Deep learning methods can be divided depending on whether they embed compounds as graphs or SMILES strings. Common graph-based approaches are Message Passing Neural Networks (MPNN), Graph Convolution Networks (GCN), Graph Attention Networks (GAT) and Attentive Fingerprints (AFP). To process string-based representations, Long Short Term Memory (LSTM) architectures, Convolutional Neural Networks (CNN) and Transformers are typically employed. Figure adapted from Ref. 62.

1.4 References

1. Beck, H.; Härter, M.; Haß, B.; Schmeck, C.; Baerfacker, L. Small Molecules and Their Impact in Drug Discovery: A Perspective on the Occasion of the 125th Anniversary of the Bayer Chemical Research Laboratory. *Drug Discov. Today* **2022**, *27* (6), 1560–1574. https://doi.org/10.1016/j.drudis.2022.02.015.
2. Lobanovska, M.; Pilla, G. Penicillin's Discovery and Antibiotic Resistance: Lessons for the Future? *Yale J. Biol. Med.* **2017**, *90* (1), 135–145.
3. Landovitz, R. J.; Scott, H.; Deeks, S. G. Prevention, Treatment and Cure of HIV Infection. *Nat. Rev. Microbiol.* **2023**, *21* (10), 657–670. https://doi.org/10.1038/s41579-023-00914-1.
4. Lichtenberg, F. R. The Effect of Pharmaceutical Innovation on Longevity: Evidence from the U.S. and 26 High-Income Countries. *Econ. Hum. Biol.* **2022**, *46*, 101124. https://doi.org/10.1016/j.ehb.2022.101124.
5. Ostwald, D. D.; Cramer, D. M.; Albu, N.; Tesch, J. The Global Economic Impact of the Pharmaceutical Industry.
6. Sun, D.; Gao, W.; Hu, H.; Zhou, S. Why 90% of Clinical Drug Development Fails and How to Improve It? *Acta Pharm. Sin. B* **2022**, *12* (7), 3049–3062. https://doi.org/10.1016/j.apsb.2022.02.002.
7. Umscheid, C. A.; Margolis, D. J.; Grossman, C. E. Key Concepts of Clinical Trials: A Narrative Review. *Postgrad. Med.* **2011**, *123* (5), 194–204. https://doi.org/10.3810/pgm.2011.09.2475.
8. Hughes, J.; Rees, S.; Kalindjian, S.; Philpott, K. Principles of Early Drug Discovery. *Br. J. Pharmacol.* **2011**, *162* (6), 1239–1249. https://doi.org/10.1111/j.1476-5381.2010.01127.x.
9. Muglia, J. J.; DiGiovanna, J. J. Phase 1 Clinical Trials. *J. Cutan. Med. Surg.* **1998**, *2* (4), 236–241. https://doi.org/10.1177/120347549800200413.
10. Iasonos, A.; O'Quigley, J. Randomised Phase 1 Clinical Trials in Oncology. *Br. J. Cancer* **2021**, *125* (7), 920–926. https://doi.org/10.1038/s41416-021-01412-y.
11. Torres-Saavedra, P. A.; Winter, K. A. An Overview of Phase 2 Clinical Trial Designs. *Int. J. Radiat. Oncol. Biol. Phys.* **2022**, *112* (1), 22–29. https://doi.org/10.1016/j.ijrobp.2021.07.1700.
12. Thall, P. F. A Review of Phase 2–3 Clinical Trial Designs. *Lifetime Data Anal.* **2008**, *14* (1), 37–53. https://doi.org/10.1007/s10985-007-9049-x.
13. Schlander, M.; Hernandez-Villafuerte, K.; Cheng, C.-Y.; Mestre-Ferrandiz, J.; Baumann, M. How Much Does It Cost to Research and Develop a New Drug? A Systematic Review and Assessment. *PharmacoEconomics* **2021**, *39* (11), 1243–1269. https://doi.org/10.1007/s40273-021-01065-y.
14. Wouters, O. J.; McKee, M.; Luyten, J. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA* **2020**, *323* (9), 844–853. https://doi.org/10.1001/jama.2020.1166.
15. Brown, D. G.; Wobst, H. J.; Kapoor, A.; Kenna, L. A.; Southall, N. T. Clinical Development Times for Innovative Drugs. *Nat. Rev. Drug Discov.* **2022**, *21* (11), 793–794. https://doi.org/10.1038/d41573-021-00190-9.
16. Dowden, H.; Munro, J. Trends in Clinical Success Rates and Therapeutic Focus. *Nat. Rev. Drug Discov.* **2019**, *18* (7), 495–496. https://doi.org/10.1038/d41573-019-00074-z.
17. Takebe, T.; Imai, R.; Ono, S. The Current Status of Drug Discovery and Development as Originated in UNITED STATES Academia: The Influence of Industrial and Academic Collaboration on Drug Discovery and Development. *Clin. Transl. Sci.* **2018**, *11* (6), 597–606. https://doi.org/10.1111/cts.12577.
18. Cohen, F. J. The Fast Track Effect. *Nat. Rev. Drug Discov.* **2004**, *3* (4), 293–294. https://doi.org/10.1038/nrd1349.
19. Chary, K. V. Expedited Drug Review Process: Fast, but Flawed. *J. Pharmacol. Pharmacother.* **2016**, *7* (2), 57–61. https://doi.org/10.4103/0976-500X.184768.

20. Disorders, F. on N. and N. S.; Policy, B. on H. S.; Medicine, I. of. Drug Development Challenges. In *Improving and Accelerating Therapeutic Development for Nervous System Disorders: Workshop Summary*; National Academies Press (US), 2014.
21. Tautermann, C. S. Current and Future Challenges in Modern Drug Discovery. *Methods Mol. Biol. Clifton NJ* **2020**, *2114*, 1–17. https://doi.org/10.1007/978-1-0716-0282-9_1.
22. Blay, V.; Tolani, B.; Ho, S. P.; Arkin, M. R. High-Throughput Screening: Today's Biochemical and Cell-Based Approaches. *Drug Discov. Today* **2020**, *25* (10), 1807–1821. https://doi.org/10.1016/j.drudis.2020.07.024.
23. Macarron, R.; Banks, M. N.; Bojanic, D.; Burns, D. J.; Cirovic, D. A.; Garyantes, T.; Green, D. V. S.; Hertzberg, R. P.; Janzen, W. P.; Paslay, J. W.; Schopfer, U.; Sittampalam, G. S. Impact of High-Throughput Screening in Biomedical Research. *Nat. Rev. Drug Discov.* **2011**, *10* (3), 188–195. https://doi.org/10.1038/nrd3368.
24. Jiménez-Luna, J.; Grisoni, F.; Weskamp, N.; Schneider, G. Artificial Intelligence in Drug Discovery: Recent Advances and Future Perspectives. *Expert Opin. Drug Discov.* **2021**, 1–11. https://doi.org/10.1080/17460441.2021.1909567.
25. Schneider, P.; Walters, W. P.; Plowright, A. T.; Sieroka, N.; Listgarten, J.; Goodnow, R. A.; Fisher, J.; Jansen, J. M.; Duca, J. S.; Rush, T. S.; Zentgraf, M.; Hill, J. E.; Krutoholow, E.; Kohler, M.; Blaney, J.; Funatsu, K.; Luebkemann, C.; Schneider, G. Rethinking Drug Design in the Artificial Intelligence Era. *Nat. Rev. Drug Discov.* **2020**, *19* (5), 353–364. https://doi.org/10.1038/s41573-019-0050-3.
26. Caraus, I.; Alsuwailem, A. A.; Nadon, R.; Makarenkov, V. Detecting and Overcoming Systematic Bias in High-Throughput Screening Technologies: A Comprehensive Review of Practical Issues and Methodological Solutions. *Brief. Bioinform.* **2015**, *16* (6), 974–986. https://doi.org/10.1093/bib/bbv004.
27. Attene-Ramos, M. S.; Austin, C. P.; Xia, M. High Throughput Screening. In *Encyclopedia of Toxicology (Third Edition)*; Wexler, P., Ed.; Academic Press: Oxford, 2014; pp 916–917. https://doi.org/10.1016/B978-0-12-386454-3.00209-8.
28. Perera, N.; Hikkaduwa Koralege, R. S. High Throughput Screening. In *Encyclopedia of Toxicology (Fourth Edition)*; Wexler, P., Ed.; Academic Press: Oxford, 2024; pp 297–301. https://doi.org/10.1016/B978-0-12-824315-2.01035-6.
29. Shelat, A. A.; Guy, R. K. The Interdependence between Screening Methods and Screening Libraries. *Curr. Opin. Chem. Biol.* **2007**, *11* (3), 244–251. https://doi.org/10.1016/j.cbpa.2007.05.003.
30. Sink, R.; Gobec, S.; Pecar, S.; Zega, A. False Positives in the Early Stages of Drug Discovery. *Curr. Med. Chem. 17* (34), 4231–4255.
31. Paricharak, S.; Méndez-Lucio, O.; Chavan Ravindranath, A.; Bender, A.; IJzerman, A. P.; Van Westen, G. J. P. Data-Driven Approaches Used for Compound Library Design, Hit Triage and Bioactivity Modelling in High-Throughput Screening. *Brief. Bioinform.* **2016**, bbw105. https://doi.org/10.1093/bib/bbw105.
32. Villar, H. O.; Hansen, M. R. Design of Chemical Libraries for Screening. *Expert Opin. Drug Discov.* **2009**, *4* (12), 1215–1220. https://doi.org/10.1517/17460440903397368.
33. Follmann, M.; Briem, H.; Steinmeyer, A.; Hillisch, A.; Schmitt, M. H.; Haning, H.; Meier, H. An Approach towards Enhancement of a Screening Library: The Next Generation Library Initiative (NGLI) at Bayer — against All Odds? *Drug Discov. Today* **2019**, *24* (3), 668–672. https://doi.org/10.1016/j.drudis.2018.12.003.
34. P. Wilson, B. A.; C. Thornburg, C.; J. Henrich, C.; Grkovic, T.; R. O'Keefe, B. Creating and Screening Natural Product Libraries. *Nat. Prod. Rep.* **2020**, *37* (7), 893–918. https://doi.org/10.1039/C9NP00068B.

35. Yang, Z.-Y.; Yang, Z.-J.; Dong, J.; Wang, L.-L.; Zhang, L.-X.; Ding, J.-J.; Ding, X.-Q.; Lu, A.-P.; Hou, T.-J.; Cao, D.-S. Structural Analysis and Identification of Colloidal Aggregators in Drug Discovery. *J. Chem. Inf. Model.* **2019**, *59* (9), 3714–3726. https://doi.org/10.1021/acs.jcim.9b00541.
36. Ganesh, A. N.; Donders, E. N.; Shoichet, B. K.; Shoichet, M. S. Colloidal Aggregation: From Screening Nuisance to Formulation Nuance. *Nano Today* **2018**, *19*, 188–200. https://doi.org/10.1016/j.nantod.2018.02.011.
37. Böcker, A.; Bonneau, P. R.; Edwards, P. J. HTS Promiscuity Analyses for Accelerating Decision Making. *SLAS Discov.* **2011**, *16* (7), 765–774. https://doi.org/10.1177/1087057111407763.
38. Rothenaigner, I.; Hadian, K. Brief Guide: Experimental Strategies for High-Quality Hit Selection from Small-Molecule Screening Campaigns. *Slas Discov.* **2021**, *26* (7), 851–854. https://doi.org/10.1177/24725552211008862.
39. Struble, T. J.; Alvarez, J. C.; Brown, S. P.; Chytil, M.; Cisar, J.; DesJarlais, R. L.; Engkvist, O.; Frank, S. A.; Greve, D. R.; Griffin, D. J.; Hou, X.; Johannes, J. W.; Kreatsoulas, C.; Lahue, B.; Mathea, M.; Mogk, G.; Nicolaou, C. A.; Palmer, A. D.; Price, D. J.; Robinson, R. I.; Salentin, S.; Xing, L.; Jaakkola, T.; Green, William. H.; Barzilay, R.; Coley, C. W.; Jensen, K. F. Current and Future Roles of Artificial Intelligence in Medicinal Chemistry Synthesis. *J. Med. Chem.* **2020**, *63* (16), 8667–8682. https://doi.org/10.1021/acs.jmedchem.9b02120.
40. David, L.; Arús-Pous, J.; Karlsson, J.; Engkvist, O.; Bjerrum, E. J.; Kogej, T.; Kriegl, J. M.; Beck, B.; Chen, H. Applications of Deep-Learning in Exploiting Large-Scale and Heterogeneous Compound Data in Industrial Pharmaceutical Research. *Front. Pharmacol.* **2019**, *10*.
41. Shen, J.; Nicolaou, C. A. Molecular Property Prediction: Recent Trends in the Era of Artificial Intelligence. *Drug Discov. Today Technol.* **2019**, *32–33*, 29–36. https://doi.org/10.1016/j.ddtec.2020.05.001.
42. Volkamer, A.; Riniker, S.; Nittinger, E.; Lanini, J.; Grisoni, F.; Evertsson, E.; Rodríguez-Pérez, R.; Schneider, N. Machine Learning for Small Molecule Drug Discovery in Academia and Industry. *Artif. Intell. Life Sci.* **2023**, *3*, 100056. https://doi.org/10.1016/j.ailsci.2022.100056.
43. Lyu, J.; Irwin, J. J.; Shoichet, B. K. Modelling the Expansion of Virtual Screening Libraries. *Nat. Chem. Biol.* **2023**, *19* (6), 712–718. https://doi.org/10.1038/s41589-022-01234-w.
44. Vázquez, J.; López, M.; Gibert, E.; Herrero, E.; Luque, F. J. Merging Ligand-Based and Structure-Based Methods in Drug Discovery: An Overview of Combined Virtual Screening Approaches. *Molecules* **2020**, *25* (20), 4723. https://doi.org/10.3390/molecules25204723.
45. Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R. K. Artificial Intelligence in Drug Discovery and Development. *Drug Discov. Today* **2021**, *26* (1), 80–93. https://doi.org/10.1016/j.drudis.2020.10.010.
46. Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackermann, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.; Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180* (4), 688-702.e13. https://doi.org/10.1016/j.cell.2020.01.021.
47. Gawriljuk, V. O.; Foil, D. H.; Puhl, A. C.; Zorn, K. M.; Lane, T. R.; Riabova, O.; Makarov, V.; Godoy, A. S.; Oliva, G.; Ekins, S. Development of Machine Learning Models and the Discovery of a New Antiviral Compound against Yellow Fever Virus. *J. Chem. Inf. Model.* **2021**, acs.jcim.1c00460. https://doi.org/10.1021/acs.jcim.1c00460.
48. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for

Property Prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237.
49. Roggia, M.; Natale, B.; Amendola, G.; Di Maro, S.; Cosconati, S. Streamlining Large Chemical Library Docking with Artificial Intelligence: The PyRMD2Dock Approach. *J. Chem. Inf. Model.* **2023**. https://doi.org/10.1021/acs.jcim.3c00647.
50. Marin, E.; Kovaleva, M.; Kadukova, M.; Mustafin, K.; Khorn, P.; Rogachev, A.; Mishin, A.; Guskov, A.; Borshchevskiy, V. Regression-Based Active Learning for Accessible Acceleration of Ultra-Large Library Docking. *J. Chem. Inf. Model.* **2023**. https://doi.org/10.1021/acs.jcim.3c01661.
51. Sivula, T.; Yetukuri, L.; Kalliokoski, T.; Käsnänen, H.; Poso, A.; Pöhner, I. Machine Learning-Boosted Docking Enables the Efficient Structure-Based Virtual Screening of Giga-Scale Enumerated Chemical Libraries. *J. Chem. Inf. Model.* **2023**, *63* (18), 5773–5783. https://doi.org/10.1021/acs.jcim.3c01239.
52. Dreiman, G. H. S.; Bictash, M.; Fish, P. V.; Griffin, L.; Svensson, F. Changing the HTS Paradigm: AI-Driven Iterative Screening for Hit Finding. *Slas Discov.* **2021**, *26* (2), 257–262. https://doi.org/10.1177/2472555220949495.
53. Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating High-Throughput Virtual Screening through Molecular Pool-Based Active Learning. *Chem. Sci.* **2021**, *12* (22), 7866–7881. https://doi.org/10.1039/D0SC06805E.
54. Aleksić, S.; Seeliger, D.; Brown, J. B. ADMET Predictability at Boehringer Ingelheim: State-of-the-Art, and Do Bigger Datasets or Algorithms Make a Difference? *Mol. Inform.* **2021**, 2100113. https://doi.org/10.1002/minf.202100113.
55. Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Noé, F.; Clevert, D.-A. Efficient Multi-Objective Molecular Optimization in a Continuous Latent Space. *Chem. Sci.* **2019**, *10* (34), 8016–8024. https://doi.org/10.1039/C9SC01928F.
56. Fromer, J. C.; Coley, C. W. Computer-Aided Multi-Objective Optimization in Small Molecule Discovery. *Patterns* **2023**, *4* (2), 100678. https://doi.org/10.1016/j.patter.2023.100678.
57. Todeschini, R.; Ballabio, D.; Cassotti, M.; Consonni, V. N3 and BNN: Two New Similarity Based Classification Methods in Comparison with Other Classifiers. *J. Chem. Inf. Model.* **2015**, *55* (11), 2365–2374. https://doi.org/10.1021/acs.jcim.5b00326.
58. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20* (3), 273–297. https://doi.org/10.1007/BF00994018.
59. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45* (1), 5–32. https://doi.org/10.1023/A:1010933404324.
60. Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models. *J. Cheminformatics* **2021**, *13* (1), 12. https://doi.org/10.1186/s13321-020-00479-8.
61. Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.
62. van Tilborg, D.; Alenicheva, A.; Grisoni, F. Exposing the Limitations of Molecular Machine Learning with Activity Cliffs. *J. Chem. Inf. Model.* **2022**, *62* (23), 5938–5951. https://doi.org/10.1021/acs.jcim.2c01073.
63. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A Comparative Analysis of Gradient Boosting Algorithms. *Artif. Intell. Rev.* **2021**, *54* (3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5.
64. Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29* (5), 1189–1232. https://doi.org/10.1214/aos/1013203451.

65. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why Do Tree-Based Models Still Outperform Deep Learning on Tabular Data? arXiv July 18, 2022. http://arxiv.org/abs/2207.08815 (accessed 2023-06-20).
66. Shwartz-Ziv, R.; Armon, A. Tabular Data: Deep Learning Is Not All You Need. *Inf. Fusion* **2022**, *81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011.
67. Lundberg, S. M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J. M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nat. Mach. Intell.* **2020**, *2* (1), 56–67. https://doi.org/10.1038/s42256-019-0138-9.

2.

Molecular property prediction in drug discovery

2.1 Fundamentals of molecular property prediction

While there is currently a wealth of different algorithms for molecular property prediction, as introduced in Chapter 1.3.2, they can generally be described under the same mathematical formalism. Furthermore, the best practices for performance evaluation are independent of the chosen modelling algorithm. As such, the aim of this section is to introduce molecular property prediction by presenting the key concepts that apply to most algorithms used for this task.

2.1.1 Quantitative Structure-Activity Relationship modelling

Molecular property prediction can be expressed with Equation 2.1:

$$y = f(x) \tag{2.1}$$

Where $x = (x^1, \dots, x^m)$ is the input molecule, expressed with a set of features m, y is the property of interest and f is the mathematical model establishing a connection between the chemical structure and the experimental response.[1] This relationship is often unknown or too complex to express analytically. As such, data-driven methods such as machine learning and deep learning provide a framework for learning a surrogate model $f_s : \mathbb{R}^m \rightarrow \mathbb{R}$ from a dataset $D = \{(x_k, y_k)\}_{k=1\dots n}$ of n molecule-property pairs, which can be used to approximate f.[1,2] These methods fall within the category of supervised learning algorithms since they require that each molecule x_k is associated with its experimental readout y_k.[3,4]

In practice, the model is obtained by adjusting the model's parameters w by minimizing the expected loss L for the training data:[3,4]

$$f_{s_w} = \arg\min_w \mathbb{E} L\left(Y, f_{s_w}(X)\right) \tag{2.2}$$

L is a loss function, which measures the discrepancy between the measured experimental properties Y for training compounds X and the value predicted by the model $f_{s_w}(X)$. In cheminformatics, these algorithms are called Quantitative Structure-Activity Relationship (QSAR) models.[1]

Depending on the property y, different types of QSAR models can be built. If y indicates a categorical response (e.g. toxic versus non-toxic), the prediction task is called classification. Instead, if y defines a numerical property (e.g. the IC50 of a molecule in a given assay), the prediction task is named regression. As such, a classification model will predict the likelihood of a compound belonging to a given class, while a regression model will output the numerical value of the property of interest associated to the input molecule.[1,2,5] However, in drug discovery the distinction between classification and regression is not as clear-cut as it is in other fields such as Computer Vision (CV) or Natural Language Processing (NLP). This is

because compound classes like “toxicity” usually stem from setting thresholds on biochemical assays that would report a numerical readout, e.g. enzyme inhibition, and considering as “toxic” every molecule that is for example below the threshold.[6,7]

Finally, molecular property values are usually measured with some degree of experimental noise.[8–10] This is problematic for the development of *in silico* predictors, since the model cannot distinguish between the true structure-activity relationship and spurious correlations from measurement uncertainty.[11,12] Because of this, the noise of the training set acts as an upper bound to the performance of a QSAR model trained on that data, since no data-driven algorithm can become more accurate than the measurements used to train it.[8,13]

2.1.2 Model evaluation

Data-driven methods tend to perform extremely well on the training dataset but may have limited predictive power on different datasets, a process known as overfitting. To have a more unbiased view of the performance of the QSAR model in perspective applications, it is customary to remove a fraction of the training dataset (e.g. 10%), named test set, which is then only used to evaluate the performance of the QSAR model on molecules not included for training.[1,2,14] There are three commonly used approaches to split the data into training and test sets:

- **Random split**: the test set is generated by randomly selecting a user-specified fraction of compounds. For classification tasks, the random sampling is usually done so that the class distribution is the same both in the training and test sets. This splitting approach is typically the least conservative in terms of performance estimation, since structural analogues of training set molecules might be present in the test set, overestimating the generalization ability of the model across the chemical space.[15]
- **Scaffold split:** the test set is generated by selecting compounds so that there is no overlap between the Bemis-Murcko scaffolds present in the training and test sets. This method is typically employed in academic benchmarks of QSAR performance, given that it allows to evaluate how well a given model generalizes to unseen regions of the chemical space. However, it might not be possible to enforce the class distribution ratio of the training and test sets to be equal.[15,16]
- **Time split:** the test set is generated by picking compounds that were analyzed in a different timeframe than the ones present in the training set, e.g. after one month. This method is considered even more challenging than scaffold splitting and it is typically used in industry benchmarks, where new measurements are routinely performed for a given assay.[14,17]

Additionally, the training and testing procedure can be repeated iteratively, generating a series of non-overlapping training and test sets. This procedure is called cross validation and allows

to obtain less biased performance metrics than the ones obtained by relying on one single test set.[5]

2.1.3 Performance metrics

Depending on whether the QSAR model is trained for classification or regression, different performance metrics are employed.

To properly evaluate classification performance, it is paramount to consider the level of class imbalance of the dataset.[18] When dealing with extremely imbalanced datasets, e.g. datasets where the number of bioactive compounds is less than 1%, correctly identifying the minority class is far more important than detecting the majority class.[19] Three metrics that are typically used to measure the performance on the minority class are recall, precision and specificity:

$$\text{Recall} = \frac{\text{TP}}{P} \tag{2.3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.4}$$

$$\text{Specificity} = \frac{\text{TN}}{N} \tag{2.5}$$

Where TP is the true positive rate, P is the number of samples belonging to the minority class, FP is the false positive rate, TN is the true negative rate and N is the number of samples belonging to the majority class. However, these metrics focus only on one aspect of the overall classification performance, e.g. the ability of the model to retrieve all bioactive compounds in a library in the case of recall, or how likely are the predicted actives to be validated experimentally in the case of precision.[20] Because of this, these metrics are typically combined to assess the classification performance as a whole.[18,21] For example, the Receiving Operator Characteristic Area Under Curve (ROC-AUC) combines recall and specificity, while the Precision-Recall Area Under Curve (PR-AUC) combines precision and recall.[18] Another popular global classification metric is the Matthews Correlation Coefficient (MCC),[22] expressed as follows:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{2.6}$$

Where TP and TN are the true positive and true negative rates, while FP and FN are the false positive and false negative rate. Additionally, for some QSAR modelling tasks, e.g. prioritizing which compounds to validate experimentally from a large library, it might be more appropriate to focus on the ability of the model to prioritize relevant compounds. Examples of metrics that measure this aspect are the Boltzmann Enhanced Discrimination of ROC (BEDROC), the enrichment factor and the top-k precision.[23]

Regarding regression, the two most popular metrics are the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE), shown in Equations 2.7 and 2.8.[5] Both measure the average discrepancy between the predicted properties and the experimental values, but RMSE punishes more harshly large errors, while MAE is more robust to outliers.[5]

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{2.7}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N} \tag{2.8}$$

2.2 Featurization methods

Converting chemical structures into a numerical format is a crucial task of cheminformatics modelling, often having a stronger impact on the performance of QSAR models than the choice of the predictive algorithm itself.[24–26] Furthermore, molecular encodings perform differently depending on the region of the chemical space to be modeled: for example, many algorithms have been developed specifically for drug-like compounds, which potentially makes them unsuitable for handling different compound classes such as natural products.[27,28] This research question is investigated in Chapter 6 of this dissertation.

Within this thesis, two classes of molecular featurization methods were investigated: molecular descriptors and fingerprinting algorithms.

2.2.1 Molecular descriptors

Molecular descriptors encompass a heterogeneous set of properties that can be quickly computed from a compound's chemical structure (Figure 2.1).[29] Depending on the QSAR modelling task, different descriptors are used to encode molecules, depending on which one correlate the most with the target property.[1,2,30] Molecular descriptors can be distinguished in three different classes: one-dimensional descriptors (1D), bidimensional descriptors (2D) and tridimensional descriptors (3D).[29]

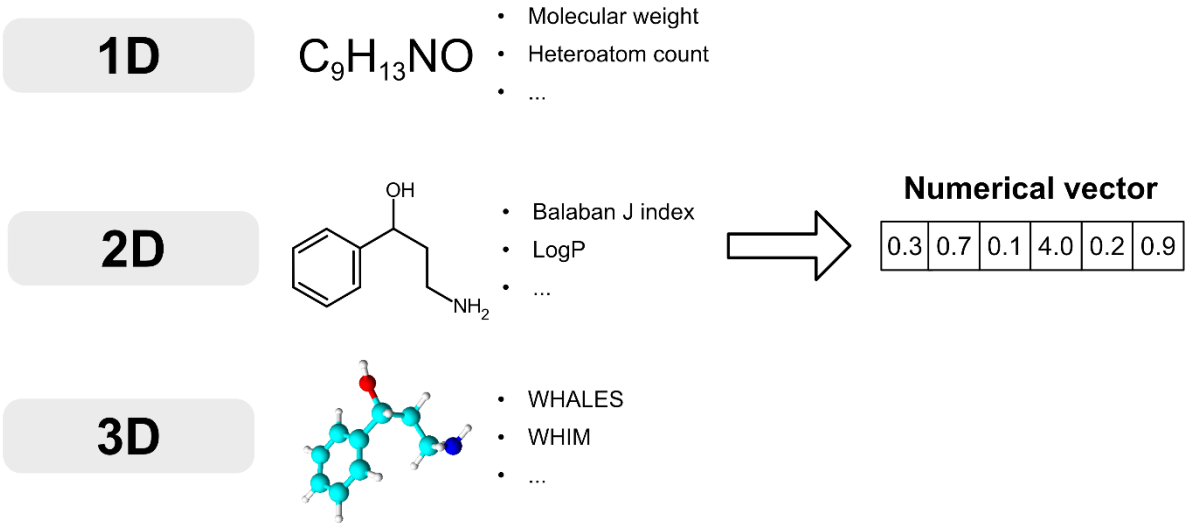


Figure 2.1 – Classification of different molecular descriptor types. 1D molecular descriptors can be computed from the chemical formula of the compound and describe the composition of the molecule. 2D molecular descriptors require the structural formula of the compound and capture information about its connectivity. Finally, 3D molecular descriptors can be computed from 3D conformers and focus on modelling the shape of the molecule.

1D molecular descriptors describe properties pertaining to the chemical constitution of a compound, irrespective of its connectivity. Examples of this class of descriptors are the molecular weight, heteroatom counts and so forth. As such, this class of descriptors can be

computed from the molecular formula of the compound, hence the "one-dimensional" denomination.[29,31]

2D molecular descriptors focus on summarizing the connectivity of the molecular graph of a given compound, meaning that they require the structural formula of the molecule to be calculated. They can range from simple counts, e.g. the number of rings in a molecule, to more complex descriptors obtained by processing the graph of the compound.[29,32,33] One example of the latter is the Balaban J index, computed as follows:[34,35]

$$J = \frac{q}{\mu + 1} \sum_{adjac(i,j)} \frac{1}{\sqrt{s_i s_j}} \qquad 2.9$$

Where q is the number of bonds, $\mu = q - n + 1$ where n is the number of atoms and s_i is the sum of the i-th row of the distance matrix of the compound. The summation is iterated only over connected atoms, and its value is scaled inversely proportional to the bond order.

Finally, 3D molecular descriptors capture the tridimensional arrangement of the atoms of a given compound. While they are more informative than 2D descriptors, they require a 3D conformer to be computed.[29,31,33] Given that often the relevant conformer for the property prediction task is unknown, they are usually computed on the most stable conformation.[29,31] This however can lead to unwanted biases if the relevant conformer differs greatly from the most stable one.[31] Two examples of 3D molecular descriptors are the Weighted Holistic Invariant Molecular (WHIM) descriptors,[36] which capture shape and symmetry information, and Weighted Holistic Atom Localization and Entity Shape (WHALES) descriptors,[37] which model the charge distribution of the conformer. The latter are computed from the Atom Centered Mahalonobis (ACM) distance matrix, calculated as follows:[37]

$$\mathrm{ACM}_{i,j} = \left(x_i - x_j\right)^T S^{-1}{}_j \left(x_i - x_j\right) \qquad 2.10$$

Where x_i is the coordinate of the i-th atom and S_j is the entry associated to the j-th atom in the atom-centered weighted covariance matrix:

$$S_j = \frac{\sum_{i=1}^{n} |\delta_i| \left(x_i - x_j\right)\left(x_i - x_j\right)^T}{\sum_{i=1}^{n} |\delta_i|} \qquad 2.11$$

Where δ_i is the electronic charge associated to the i-th atom.

2.2.2 Fingerprints

Fingerprints convert molecular structures into bit vectors by decomposing their chemical graph with a predefined algorithm.[38,39] Each bit encodes whether a given molecular motif is present or absent in the target molecule.[1,2] Fingerprinting algorithms can be classified into five families (Figure 2.2):[40]

- Substructure-based
- Path-based
- Circular
- Pharmacophore
- String-based

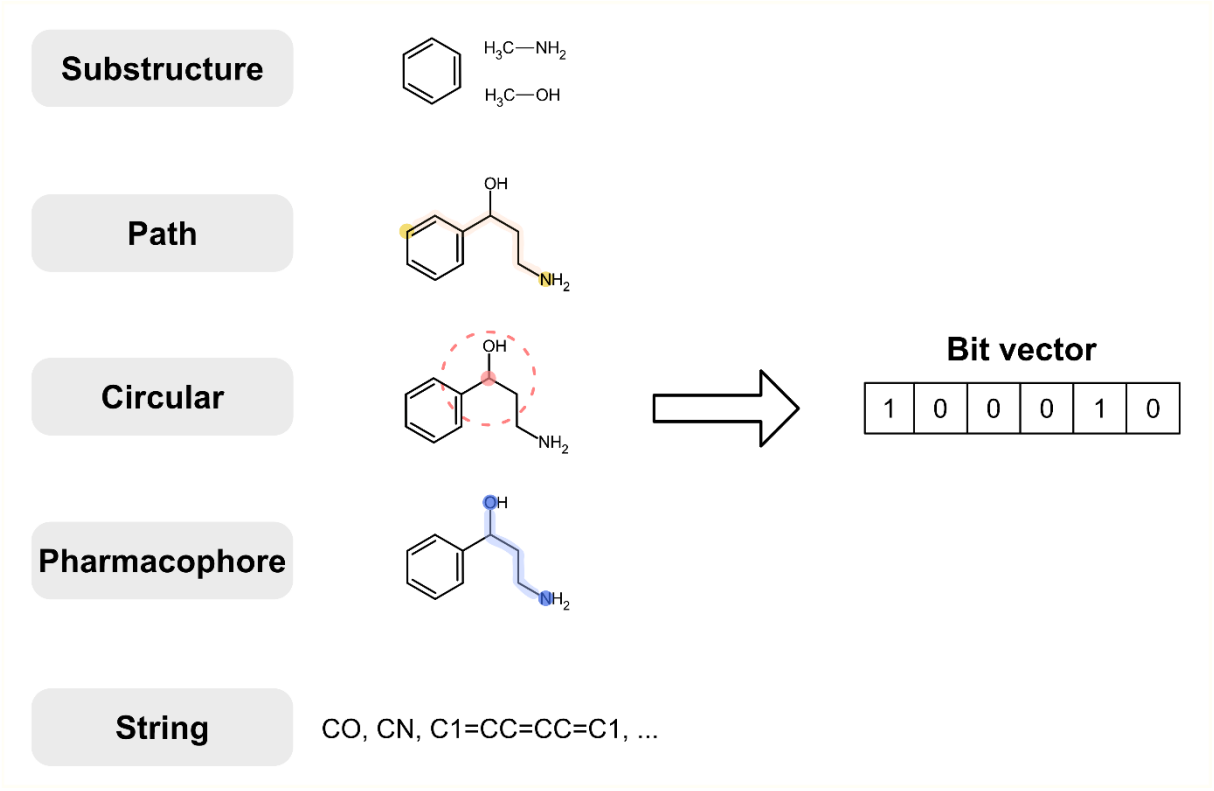


Figure 2.2 – Classification of different molecular fingerprint types. Substructure-based approaches use expert-defined fragments to represent compounds. Path-based and circular algorithms process the topology of the chemical graph of the compound. Pharmacophore fingerprints focus on the arrangement of pharmacophores in the compound structure. String algorithms identify fragments by processing the SMILES representation of the molecule.

Substructure-based fingerprinting algorithms are fundamentally a set of functional groups and molecular fragments.[39] These are typically manually curated by medicinal chemists and cheminformaticians, and usually include the most relevant chemical features for pharmacological applications.[39,41] To generate substructure-based fingerprints, the algorithm checks iteratively if each motif is present or absent in the target compound, returning a bit vector of the same dimensionality as the number of fragments evaluated. Examples of this class of fingerprints are the MACCS keys and the PubChem fingerprints.[27,39,42]

Path-based fingerprints have two key differences from substructure-based algorithms. First, they do not have a predefined dimensionality. Second, due to how they are computed, each bit does not necessarily encode the same molecular fragment for two different compounds. These encodings are generated by enumerating the unique paths between two vertexes of the molecular graph, which are then stored into a bit vector of fixed size via a a hash function.[39,43,44] For example, the All-Shortest Paths (ASP) fingerprint is computed as follows:[39]

$$\mathrm{ASP} = \bigcup_{i=1}^{n} \{\mathrm{DFS}(a_i, d), |p_{ij}| = t_{ij}\} \quad 2.12$$

Where n is the number of atoms in the molecule, DFS indicates a depth-first search of the molecular graph from atom a_i up to a number of bonds d, p_{ij} indicates the number of bonds between the i-th and j-th atom and t_{ij} denotes the minimum path possible to connect those two atoms. In practice, the algorithm iterates over each atom, collecting the shortest path connecting the root atom to another atom up to a bond distance d, ensuring that for a given atom pair only the shortest path is preserved. Then, all the unique paths obtained this way are hashed into a fixed-size representation. By changing the criteria for path inclusion and calculation, different path-based fingerprints can be computed, such as the Atom Pair (AP) fingerprint and the DFS fingerprint.[39,45]

Circular fingerprints work similarly to path-based algorithms, in the sense that they dynamically compute the fragments for each compound and rely on hashing to convert the set of chemical motifs to a fixed-size vector, but encode progressively larger atomic radial neighborhoods rather than paths along the chemical graph.[39] One key aspect of circular fingerprint computation is the concept of atomic identifiers. To generate numerical representations of the fragments for a given molecule, it is necessary to aggregate information from all the atoms in the radial neighborhood. This operation in turn requires a numerical encoding for the atoms, which is usually obtained by choosing a set of atomic properties to represent atoms, such as the atomic number, its hybridization state and so forth. These properties are called atom identifiers and strongly impact the performance of the molecular fingerprint.[39] For example, Extended Connectivity Fingerprints (ECFP) and Functional Class Fingerprints (FCFP) only differ in terms of atom identifies, but show different behavior in terms of similarity searching and QSAR modelling.[46]

Pharmacophore fingerprints use similar algorithms as the ones found in path-based fingerprints, but focus on encoding the arrangement of pharmacophoric points in the molecule, such as hydrogen bond donors and acceptors.[39,47] Therefore, these fingerprints aim to capture how a given compound interacts with its environment, e.g. with the binding site of a protein, rather than accurately encoding the chemical structure of the molecule of interest. Examples of this class of fingerprints are Pharmacophoric Pairs (PP2).[48]

$$\mathrm{PP2} = \bigcup_{i,j}^{n} P_i \oplus t_{ij} \oplus P_j \oplus t_{ji} \quad 2.13$$

Where P_i denotes the set of valid pharmacophore properties for the i-th atom and t_{ij} indicates the shortest path between the i-th and j-th atom. The calculation of pairwise atomic distances along the graph is repeated for each pharmacophore property.

Finally, string-based fingerprints generate molecular encodings by decomposing the SMILES string of the compound, rather than operating on its graph representation.[49] Examples of this class of encodings are MinHashed Fingerprints (MHFP),[50] a SMILES-based variation of ECFP, and MinHashed Atom Pair Fingerprints (MAP4),[28] which uses SMILES in combination with a path-based algorithm.

2.3 Gradient boosting machines

Gradient boosting is a powerful and efficient tree ensemble algorithm that has found widespread success in a variety of fields and is becoming increasingly popular for QSAR modelling.[24,51–53] In this chapter, the general mechanism behind GBMs is introduced, as well as its three most popular implementations: XGBoost, LightGBM and CatBoost.

While these algorithms are often used interchangeably in the cheminformatics community, they have substantial algorithmic differences, which translate in different performance and computational efficiency.[54] Chapter 4 of this dissertation benchmarks XGBoost, LightGBM and CatBoost in terms of molecular property prediction to determine the best practices for their use and optimization.

2.3.1 Constructing ensembles via boosting

GBM is an ensemble algorithm, meaning that it aims to obtain a highly performant model by aggregating a set of weak predictors.[55] The resulting QSAR model can be expressed as follows:[56]

$$\mathrm{GBM}(x) = \sum_{i=1}^{k} \sigma f_i(x) \qquad 2.14$$

Where x is e.g. the set of molecular descriptors for the input compound, σ is a regularization parameter called learning rate and f_i is the i-th predictor in an ensemble with k predictors. While this procedure can be applied using any algorithm for learning f_i, GBMs are typically trained by ensembling regression trees. This is because regression trees are non-parametric, can handle missing and categorical data and are extremely computationally efficient, making it possible to train large ensembles with higher predictive performance.[55,56]

Regression trees are constituted by nodes and leaves.[57–59] Nodes are used to rout samples within the tree structure, each with its own binary decision, e.g. whether the compound has a molecular weight above or below 350 Da. The nodes are structured in a hierarchical fashion, with each new node further splitting the compounds reaching its parent node within the tree. The terminal nodes are called leaves, which use their weights to compute the sample's prediction.[57–59] As such, predictions in regression trees are computed by routing the molecule within the tree until a leaf is reached, returning the leaf's weight as the output. The regression tree structure, splits and weights are greedily optimized by maximizing a quality-of-fit criterion such as variance reduction.[57–59] A regression tree example for solubility prediction is shown in Figure 2.3.

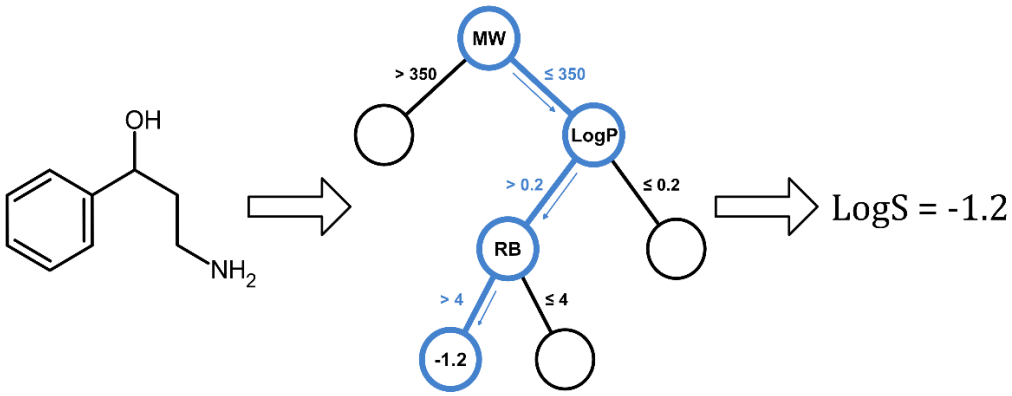


Figure 2.3 – Example of a regression tree for solubility (LogS) prediction. The target molecule has a molecular weight (MW) less than 350 Da, an octanol-water partition coefficient (LogP) larger than 0.2 and less than 4 rotatable bonds (RB). The compound is therefore routed to the corresponding leaf node, and the predicted solubility corresponds to the weight of the final node.

To construct an effective ensemble of regression trees, the GBM algorithm fits the weak predictors in a sequence, so that each new member of the ensemble compensates for the errors of the model in the prior iteration.[55] As such, given a loss function L, at each iteration i the new f_i with configuration w is computed by minimizing the following objective:[60]

$$f_{w_i} = \arg\min_w \mathbb{E}\left(\frac{-\partial L(y, p_{i-1})}{\partial \mathrm{GBM}_{i-1}} - p_{i-1}\right) \qquad 2.15$$

Where y are the measured molecular properties to predict and p_{i-1} are the predictions of the GBM ensemble at the prior iteration. Given that each new predictor is fit according to the gradient of the loss function at the previous step, this procedure can be understood as performing gradient descent in function space instead of weight space (Figure 2.4).[55]

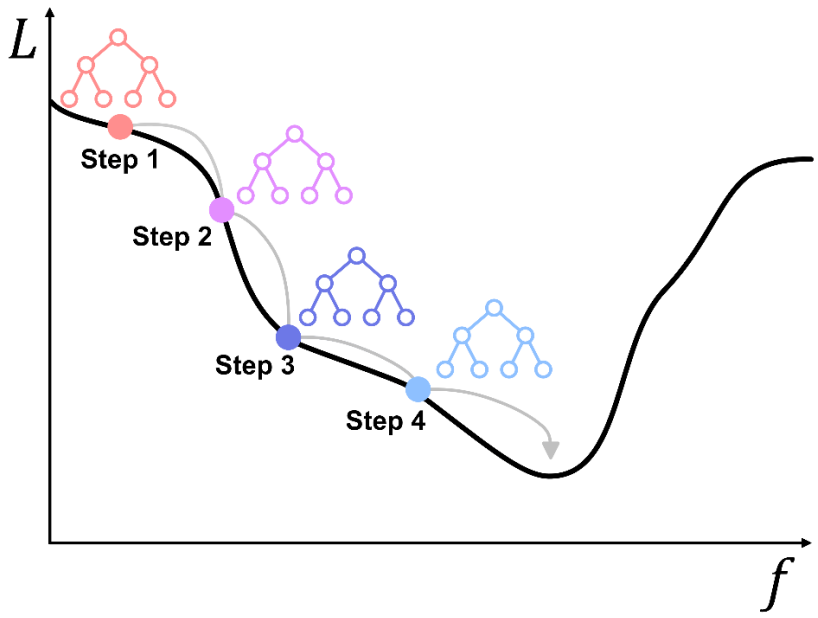


Figure 2.4 – Gradient descent in function space, according to the Gradient Boosting Machine algorithm. At each optimization step, a new regression tree is fit according to the residuals of the previous model, progressively reducing the loss of the ensemble.

2.3.2 XGBoost

The two main contributions of XGBoost to the GBM algorithm are the introduction of a regularized learning objective and the use of Newton descent instead of gradient descent for model training.[56]

The new learning objective adds L2 regularization to the loss of the GBM ensemble, as shown in Equation 2.16:[56]

$$L_{\emptyset} = \sum_{i=1}^{I} L(y_i, p_i) + \sum_{k=1}^{K} \gamma T_k + \frac{1}{2}\lambda \|w_k\|^2 \qquad 2.16$$

Where y_i is the molecular property of the i-th compound, p_i is the i-th model prediction, T_k is the number of leaves in the k-th tree, γ is a hyperparameter penalizing tree depth, w_k are the leaf weights of the k-th tree and λ is the L2 regularization hyperparameter. This modification forces the GBM algorithm to learn shallower trees with smaller weights, reducing the risk of overfitting on the training data.[56]

Using Newton descent allows faster optimization of the learning objective by leveraging both the gradients and the Hessians of the loss function.[5] As such, the equations to determine the split gain L_{split} and optimal leaf weights $w_j$are changed as follows:

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \qquad 2.17$$

$$L_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \qquad 2.18$$

Where g_i and h_i are the$i$-th gradient and Hessian, letting $I = I_L \cup I_R$and I_j indicating the set of samples being routed to the j-th leaf.

2.3.3 LightGBM

LightGBM includes both the regularized learning objective and Newton descent optimization introduced by XGBoost, but further improves the computational efficiency of GBMs by adding Exclusive Feature Bundling (EFB) and Gradient-based One Sided Sampling (GOSS).[61]

EFB is a variable selection algorithm which enables GBM to identify which input features are approximately mutually exclusive, e.g. a pair of binary fingerprint bits that never have a value of 0 simultaneously. Given a set of molecular features M and two mutually exclusive features m_i and m_j, if $m_i \in M$ then m_j is redundant, given that its values can be already inferred from m_i. Grouping together all mutually exclusive features and all approximately exclusive features

up to an approximation error e allows to reduce the dimensionality of the inputs of the GBM algorithm, thus reducing the computational time required to train the model.[61]

GOSS is a modification of the stochastic variation of the GBM algorithm, where at each boosting iteration only a fraction of the training samples is used to train the i-th regression tree. Instead of sampling randomly, GOSS prioritizes samples with higher gradients while upscaling the contribution of sampled training instances with lower gradients. This procedure avoids changing the gradient distribution of the training set while selecting only the most relevant samples, leading to an increase in computational efficiency without compromising on accuracy.[61]

Finally, LightGBM uses a different type of regression trees to construct the GBM ensemble, named depth-first trees. XGBoost and other GBM implementations use breadth-first trees, where new nodes are added so that the maximum depth of the tree is constant across all branches. The regression trees used in LightGBM instead are grown by splitting the most optimal node at each iteration, potentially leading to depth imbalance between tree branches. This approach is computationally faster but can lead to deeper trees which can have worse generalization.[61]

2.3.4 CatBoost

CatBoost builds up from the XGBoost implementation by adding two additional features: a novel Target Statistics (TS) algorithm for handling categorical data and the ordered boosting, a variation of the procedure typically employed for GBM training.[60]

The TS algorithm used by CatBoost enables it to produce better embeddings for categorical variables by avoiding overfitting on the training data distribution. However, given that typically molecular representations do not use categorical variables, this is of little utility for QSAR modelling.[60]

Ordered boosting deals with the issue of prediction shift, whereas the ensemble grows in size, the distribution of the gradients of the training samples begins to shift away from the one of test samples, as a consequence of fitting multiple times on the same training instances.[55,62,63] This leads to a loss of generalization for the GBM model. To tackle this, at each iteration ordered boosting computes g_i and h_i by using a tree trained on a dataset permutation D_i where $i \notin D_i$.[60]

Finally, CatBoost uses oblique trees, a variant of regression trees where at a given depth level, all nodes use the same variable and threshold to compute the split. Enforcing this constraint acts as a regularization technique on the tree structure, producing less expressive but potentially less overfitting regression trees than depth-first or breadth-first trees.[60,64]

2.4 Imbalanced classification

One of the main challenges of modelling classification dataset is class imbalance, where one class is disproportionately more prevalent in the training data than the others.[18] This bias tends to skew the classifier towards ignoring the minority class, leading to unsatisfactory predictive performance.[19,65] This issue is even more crucial when modelling HTS bioactivity data, where extreme class imbalance is the norm and the identification of the minority class is usually the priority for virtual screening campaigns.[16,19] To tackle this issue, many approaches have been developed. This chapter provides a brief overview of the most popular methods to combat class imbalance for molecular property prediction.

2.4.1 Resampling

A common approach for imbalanced classification in the context of molecular property prediction is resampling, consisting of manipulating the training data so that the class distribution is more balanced (Figure 2.5).[18,66–68] This can be done in two ways: undersampling and oversampling.[18] In the case of undersampling, a fraction of majority class compounds are removed from the training data, either randomly or according to some heuristic.[67] For example, Tomek's links suggests the removal of majority class samples that have a minority class instance as its closest neighbor, in an effort to boost the separation between the two classes.[69] The same approach can be expanded to use the top-k nearest neighbors to further smooth the class boundary.[69] In the case of oversampling, synthetic minority class samples are added to the training data, for example by randomly duplicating existing training instances.[67] Alternatively, artificial samples are typically generated by interpolating between minority class instances, either selected at random, e.g. in the case of Synthetic Minority Oversampling Technique (SMOTE),[70] or by focusing on samples close to the class boundary, as done by Adaptive Synthetic Oversampling (ADASYN).[71]

Unfortunately, both undersampling and oversampling have significant disadvantages. Undersampling, especially when dealing with extreme class imbalance, forces the exclusion of a considerable number of molecules from the training data, limiting the applicability domain of the QSAR model. On the other hand, oversampling increases training time and can lead to overfitting due to the presence of synthetic data.[18]

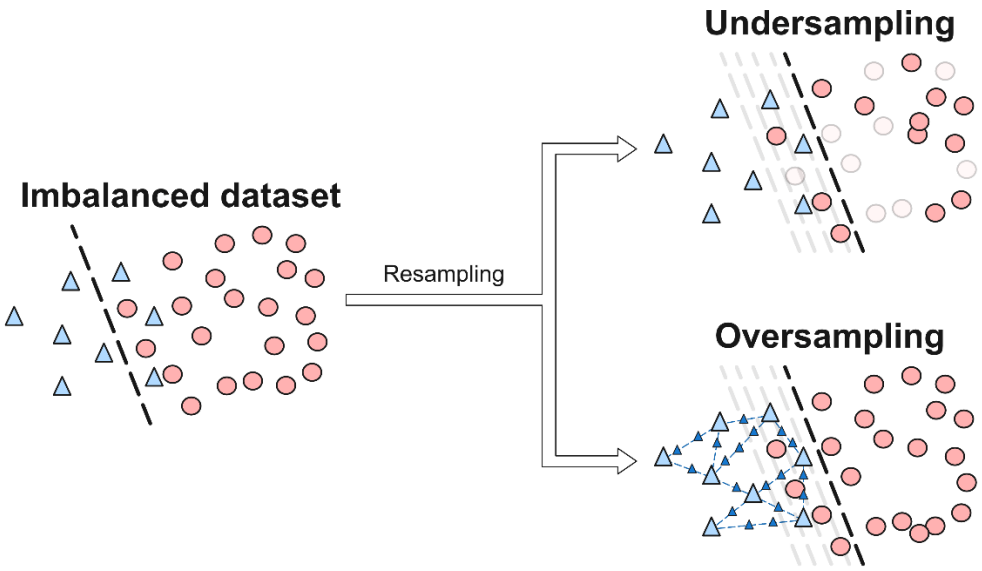


Figure 2.5 – Example of resampling approaches for modelling imbalanced classification datasets. In the undersampling case, training instances belonging to the majority class are removed at random. In the oversampling scenario, new synthetic minority class samples are generated by interpolating between instances of the minority class, according to the Synthetic Minority Oversampling Technique (SMOTE) algorithm. In both cases, the numerical imbalance between majority and minority classes is reduced, thus leading to a more optimal decision boundary for the identification of the minority class.

2.4.2 Custom loss functions

Because of the issues of resampling approaches, in recent years adjusting the learning objective has become a popular solution for handling class imbalance, especially for classification in Computer Vision, where datasets often have extremely skewed class distributions.[72,73] These approaches generally revolve around modifying the cross-entropy loss to account for the higher misclassification cost on the minority class (Figure 2.6).[67] The simplest example of this approach is weighted cross-entropy, where weights are added to the class loss terms so that errors on the minority class are weighted by the inverse of the class ratio.[21] Recent research however has shown that this approach is suboptimal and that other modifications lead to better performance, such as Focal loss and Logit-Adjusted loss.[72,74] The former forces the model to focus on hard-to-classify samples, while the latter biases the cross-entropy loss by incorporating a logit shift proportional to the class frequencies.

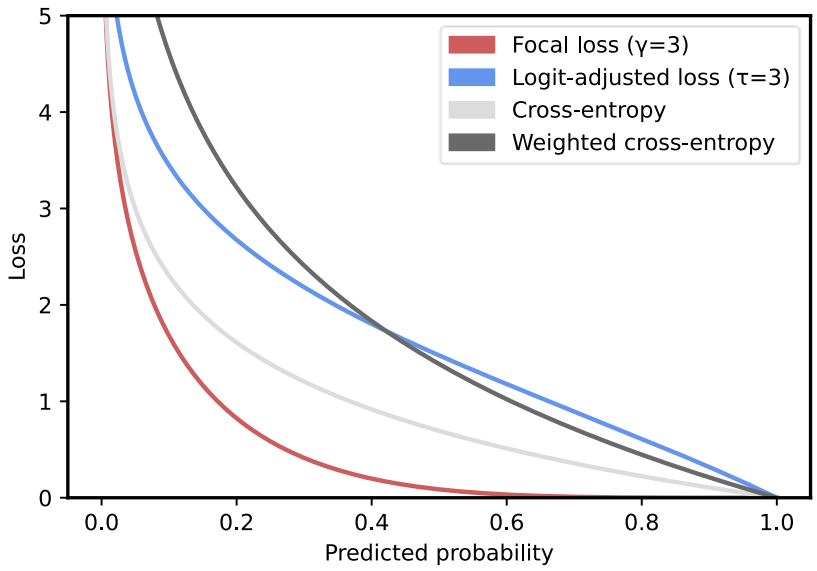


Figure 2.6 – Behaviour of different loss functions when modelling an imbalanced classification dataset, where the minority class (denoted with label "1") constitutes only a third of the total number of samples. Given a sample belonging to class "1", all losses decrease as the probability predicted by the model approaches 1.0. Focal loss already decays to 0 at around 0.6 predicted probability, thus limiting the benefit the model gains from further improving the prediction once the sample is classified reasonably well. Logit-adjusted loss and weighted cross-entropy both upscale the loss associated to samples belonging to class "1", albeit with different magnitudes depending on the predicted probability. Canonical cross-entropy is a middle ground between Focal loss, Logit-Adjusted loss and weighted cross-entropy.

While these approaches can theoretically be used by any classifier for which the loss function can be changed, e.g. GBMs, using custom loss functions has been limited so far to neural network training. Additionally, the use of custom loss functions beyond weighted cross entropy hasn't been popularized yet for cheminformatics applications. This research gap is investigated in Chapter 3.

2.4.3 Thresholding

Another attractive option for tackling class imbalance is thresholding, a post-training adjustment to the outputs of the classifier to improve the reliability of class labels.[21,67] One example of this method for molecular property prediction is Generalized Threshold Shifting (GHOST), which can improve the classification performance of a QSAR model by determining the optimal probability threshold to distinguish between active and inactive molecules.[65] While this approach is computationally lightweight and universal, in the sense that it can be used in conjunction with any classification algorithm, it does not push the classifier to learn a better class boundary. Additionally, it only improves imbalanced classification performance for metrics requiring class labels, e.g. MCC, while it does not have an impact on figures of merit requiring probabilities such as ROC-AUC or PR-AUC.

2.5 Data valuation for QSAR models

Data valuation is a new field of machine learning research investigating which training instance is most influential on the behavior of a data-driven model (Figure 2.7).[75,76] While this subject has numerous applications, such as providing instance-based explanations or deepening our understanding of how AI algorithms learn, one key finding is that data valuation can help identify mislabeled samples in classification tasks.[77,78] This is especially important in the context of HTS campaigns, where false positives are commonplace.[79] While there is a wealth of *in silico* approaches for detecting false actives in HTS data, they all make assumptions on the interference mechanism, chemical space or assay technology used during the screen, making their applicability limited to specific scenarios.[80–83] Data valuation algorithms instead can theoretically detect any type of interferent, meaning that they can then also correctly prioritize hits with the highest chance of being true positives. As such, this class of algorithms is uniquely positioned to boost efficient hit prioritization of HTS campaigns.

In this chapter, the general theory behind data valuation for machine learning models is presented. In chapter 5, a new algorithm for data valuation for GBMs is presented and it is benchmarked on a wide variety of HTS datasets on both false positive and true positive detection.

2.5.1 From influence functions to gradient tracing

An idealized notion of influence could be expressed as the impact a given training sample z has on the loss associated to a test instance z'. Formally, given a predictor with weights w and minimizing a loss $L = \sum_{i=1}^{I} l(w, z_i)$, trained with gradient descent using one sample at a time, the importance of z on z' can be expressed as follows:[77]

$$\text{Importance}(z, z') = \sum_{t:z_t=z} l(w_t, z') - l(w_{t+1}, z') \qquad 2.19$$

Where t denotes a gradient descent step performed after processing z and w_t indicates the weight configuration of the predictor at the t-th step. Stemming from this, to compute the influence z has on the dataset Z, it is possible to iterate equation 2.19 on all $z' \in Z$. The fundamental issue with this idealized notion of influence is that it involves retraining the model for each instance in the training dataset, which quickly becomes unfeasible as the dataset size and algorithmic complexity increase.[77] To tackle this issue, a number of alternative approaches have been suggested.

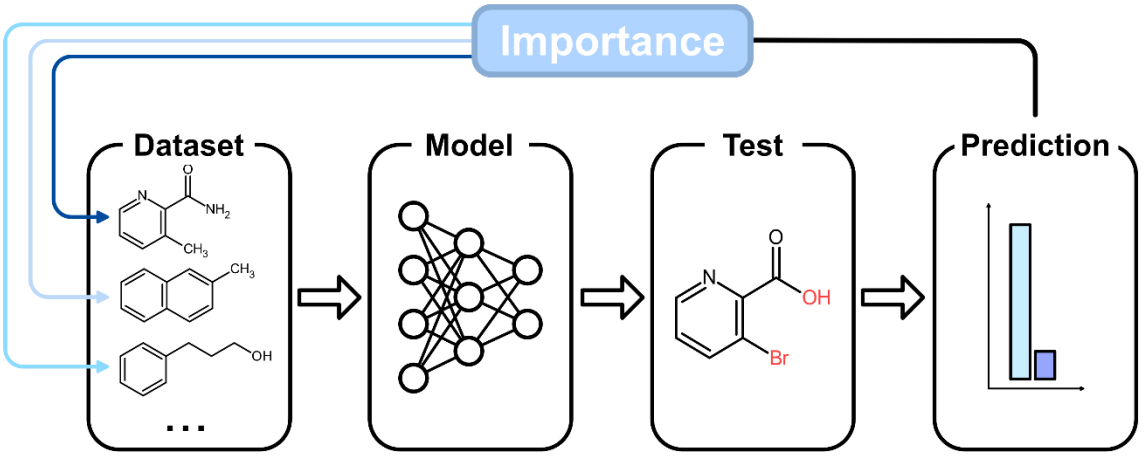


Figure 2.7 – General data valuation workflow for molecular property prediction models. First, a machine-learning or deep-learning model is obtained by fitting the training dataset. Then, it is possible to compute which training instances impact the prediction of a new test compound the most by using a sample importance estimation algorithm. Intuitively, the first molecule from the top in the training dataset has higher importance for the test compound given their structural similarity.

The first approach employed to compute sample importance used approximations of influence functions (IF).[84] In short, using the same notation as above, sample importance can be expressed as follows:

$$\text{Importance}_{\text{IF}}(z) = -\frac{1}{n}\frac{\partial w_{\varepsilon,z}}{\partial \varepsilon}\bigg|_{\varepsilon=0} \tag{2.20}$$

Where n is the number of samples in Z and ε is an arbitrarily small perturbation of the weights of the model. In practice, while Equation 2.20 allows to avoid retraining the model n times, it involves calculating the inverse of the Hessian for the loss function, which can be prohibitively expensive for large models. Additionally, IF approaches have generally performed poorly in noisy label detection benchmarks.[75,77,85]

A popular alternative to influence functions is Data Shapley (DS).[75,85] Originally from game theory, this approach estimates the importance of a given training point by framing the training process as a cooperative game, where the gains from training (in this context, the performance of the model U) have to be distributed fairly among the samples in the training set.[86] This is done by evaluating the marginal contribution of z to all possible subsets of Z without it:

$$\text{Importance}_{\text{DS}}(z) = \frac{1}{n}\sum_{S\subseteq Z\setminus\{z\}}\frac{1}{\binom{n-1}{n_S}}[U(S\cup\{z\}) - U(S)] \tag{2.21}$$

Where n_S is the number of samples in the partition S. While there is a number of heuristics for faster computation of Equation 2.21, such as Monte Carlo approximations[75] and replacing the underlying classifier with a KNN model,[87,88] DS remains difficult to scale effectively past dataset sizes above 1000 samples.[89]

Finally, inspection of gradients while training deep learning models has been suggested as an effective method for estimating sample importance in neural networks.[77] This method, named TracIn, can be formally defined as follows:

$$\text{Importance}_{\text{TracIn}}(z, z') = \sum_{i=1}^{k} \sigma_i * \nabla l(w_i, z) * \nabla l(w_i, z) \qquad 2.22$$

Where k is the number of weight snapshots taken during training and σ_i is the learning rate of the network at the i-th snapshot. While this method has achieved remarkable performance for noisy label detection, it holds some disadvantages. First, similarly to Equation 2.19, in order to compute the global influence of sample z it is necessary to loop Equation 2.22 for each $z' \in Z$, which can be particularly computationally intensive. Second, the calculation of all partial derivatives of the loss function $\nabla l(w_i, z)$ can be very costly for large networks, and choosing which layer to limit the computation of the derivatives on is not straightforward.[77,78] Finally, TracIn is intrinsically limited to deep learning models, which hampers its versatility for data valuation.

Concerning GBM, little work has been done in terms of data valuation so far. The implementation of IF inspired or TracIn-like algorithms is not straightforward, given that this class of algorithms has both weights and decision splits, making derivative-based approaches complicated. CatBoost offers a leave-one-out re-training approximation, named Object Importance, but it can only consider changes on the leaf weights, while the rest of the tree structure is kept constant.[90] Finally, using DS-like algorithms could potentially work, but it would likely be computationally unfeasible, given that its application to simpler methods such as KNN already struggles to scale to large datasets.

2.6 References

1. Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; Isayev, O.; Curtalolo, S.; Fourches, D.; Cohen, Y.; Aspuru-Guzik, A.; Winkler, D. A.; Agrafiotis, D.; Cherkasov, A.; Tropsha, A. QSAR without Borders. *Chem. Soc. Rev.* **2020**, *49* (11), 3525–3564. https://doi.org/10.1039/D0CS00098A.
2. Neves, B. J.; Braga, R. C.; Melo-Filho, C. C.; Moreira-Filho, J. T.; Muratov, E. N.; Andrade, C. H. QSAR-Based Virtual Screening: Advances and Applications in Drug Discovery. *Front. Pharmacol.* **2018**, *9*.
3. Mendonça, M. O. K.; Netto, S. L.; Diniz, P. S. R.; Theodoridis, S. Chapter 13 - Machine Learning: Review and Trends. In *Signal Processing and Machine Learning Theory*; Diniz, P. S. R., Ed.; Academic Press, 2024; pp 869–959. https://doi.org/10.1016/B978-0-32-391772-8.00019-3.
4. Kotsiantis, S. B. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*; IOS Press: NLD, 2007; pp 3–24.
5. Bishop, C. M. *Pattern Recognition and Machine Learning*; Information science and statistics; Springer: New York, 2006.
6. Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chem. Sci.* **2018**, *9* (24), 5441–5451. https://doi.org/10.1039/C8SC00148K.
7. Lenselink, E. B.; ten Dijke, N.; Bongers, B.; Papadatos, G.; van Vlijmen, H. W. T.; Kowalczyk, W.; IJzerman, A. P.; van Westen, G. J. P. Beyond the Hype: Deep Neural Networks Outperform Established Methods Using a ChEMBL Bioactivity Benchmark Set. *J. Cheminformatics* **2017**, *9* (1), 45. https://doi.org/10.1186/s13321-017-0232-0.
8. Mervin, L. H.; Trapotsi, M.-A.; Afzal, A. M.; Barrett, I. P.; Bender, A.; Engkvist, O. Probabilistic Random Forest Improves Bioactivity Predictions Close to the Classification Threshold by Taking into Account Experimental Uncertainty. *J. Cheminformatics* **2021**, *13* (1), 62. https://doi.org/10.1186/s13321-021-00539-7.
9. Kramer, C.; Kalliokoski, T.; Gedeck, P.; Vulpetti, A. The Experimental Uncertainty of Heterogeneous Public K_i Data. *J. Med. Chem.* **2012**, *55* (11), 5165–5173. https://doi.org/10.1021/jm300131x.
10. Kalliokoski, T.; Kramer, C.; Vulpetti, A.; Gedeck, P. Comparability of Mixed IC50 Data – A Statistical Analysis. *PLoS ONE* **2013**, *8* (4), e61007. https://doi.org/10.1371/journal.pone.0061007.
11. Aldeghi, M.; Graff, D. E.; Frey, N.; Morrone, J. A.; Pyzer-Knapp, E. O.; Jordan, K. E.; Coley, C. W. Roughness of Molecular Property Landscapes and Its Impact on Modellability. *J. Chem. Inf. Model.* **2022**, *62* (19), 4660–4671. https://doi.org/10.1021/acs.jcim.2c00903.
12. Luque Ruiz, I.; Gómez-Nieto, M. Á. Regression Modelability Index: A New Index for Prediction of the Modelability of Data Sets in the Development of QSAR Regression Models. *J. Chem. Inf. Model.* **2018**, *58* (10), 2069–2084. https://doi.org/10.1021/acs.jcim.8b00313.
13. Schaduangrat, N.; Lampa, S.; Simeon, S.; Gleeson, M. P.; Spjuth, O.; Nantasenamat, C. Towards Reproducible Computational Drug Discovery. *J. Cheminformatics* **2020**, *12* (1), 9. https://doi.org/10.1186/s13321-020-0408-x.
14. Volkamer, A.; Riniker, S.; Nittinger, E.; Lanini, J.; Grisoni, F.; Evertsson, E.; Rodríguez-Pérez, R.; Schneider, N. Machine Learning for Small Molecule Drug Discovery in Academia and Industry. *Artif. Intell. Life Sci.* **2023**, *3*, 100056. https://doi.org/10.1016/j.ailsci.2022.100056.

15. Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.
16. Keshavarzi Arshadi, A.; Salem, M.; Firouzbakht, A.; Yuan, J. S. MolData, a Molecular Benchmark for Disease and Target Based Machine Learning. *J. Cheminformatics* **2022**, *14* (1), 10. https://doi.org/10.1186/s13321-022-00590-y.
17. Sheridan, R. P. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model.* **2013**, *53* (4), 783–790. https://doi.org/10.1021/ci400084k.
18. Feng, Y.; Zhou, M.; Tong, X. Imbalanced Classification: A Paradigm-Based Review. arXiv June 30, 2021. http://arxiv.org/abs/2002.04592 (accessed 2022-10-10).
19. Korkmaz, S. Deep Learning-Based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* **2020**, *60* (9), 4180–4190. https://doi.org/10.1021/acs.jcim.9b01162.
20. Ballabio, D.; Grisoni, F.; Todeschini, R. Multivariate Comparison of Classification Performance Measures. *Chemom. Intell. Lab. Syst.* **2018**, *174*, 33–44. https://doi.org/10.1016/j.chemolab.2017.12.004.
21. Sun, Y.; Wong, A. K. C.; Kamel, M. S. Classification of Imbalanced Data: a Review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23* (04), 687–719. https://doi.org/10.1142/S0218001409007326.
22. Chicco, D.; Jurman, G. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC Genomics* **2020**, *21* (1), 6. https://doi.org/10.1186/s12864-019-6413-7.
23. Zhao, W.; Hevener, K. E.; White, S. W.; Lee, R. E.; Boyett, J. M. A Statistical Framework to Evaluate Virtual Screening. *BMC Bioinformatics* **2009**, *10* (1), 225. https://doi.org/10.1186/1471-2105-10-225.
24. van Tilborg, D.; Alenicheva, A.; Grisoni, F. Exposing the Limitations of Molecular Machine Learning with Activity Cliffs. *J. Chem. Inf. Model.* **2022**, *62* (23), 5938–5951. https://doi.org/10.1021/acs.jcim.2c01073.
25. Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning Continuous and Data-Driven Molecular Descriptors by Translating Equivalent Chemical Representations. *Chem. Sci.* **2019**, *10* (6), 1692–1701. https://doi.org/10.1039/C8SC04175J.
26. Capecchi, A.; Reymond, J.-L. Classifying Natural Products from Plants, Fungi or Bacteria Using the COCONUT Database and Machine Learning. *J. Cheminformatics* **2021**, *13* (1), 82. https://doi.org/10.1186/s13321-021-00559-3.
27. Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (6), 1273–1280. https://doi.org/10.1021/ci010132r.
28. Capecchi, A.; Probst, D.; Reymond, J.-L. One Molecular Fingerprint to Rule Them All: Drugs, Biomolecules, and the Metabolome. *J. Cheminformatics* **2020**, *12* (1), 43. https://doi.org/10.1186/s13321-020-00445-4.
29. Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*, 1st ed.; Methods and Principles in Medicinal Chemistry; Wiley, 2000. https://doi.org/10.1002/9783527613106.
30. Chuang, K. V.; Gunsalus, L. M.; Keiser, M. J. Learning Molecular Representations for Medicinal Chemistry: Miniperspective. *J. Med. Chem.* **2020**, *63* (16), 8705–8722. https://doi.org/10.1021/acs.jmedchem.0c00385.
31. Bahia, M. S.; Kaspi, O.; Touitou, M.; Binayev, I.; Dhail, S.; Spiegel, J.; Khazanov, N.; Yosipof, A.; Senderowitz, H. A Comparison between 2D and 3D Descriptors in QSAR Modelling Based on Bio-active Conformations. *Mol. Inform.* **2023**, *42* (4), 2200186. https://doi.org/10.1002/minf.202200186.
32. Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T. Mordred: A Molecular Descriptor Calculator. *J. Cheminformatics* **2018**, *10* (1), 4. https://doi.org/10.1186/s13321-018-0258-y.

33. Wigh, D. S.; Goodman, J. M.; Lapkin, A. A. A Review of Molecular Representation in the Age of Machine Learning. *WIREs Comput. Mol. Sci.* **2022**, *12* (5), e1603. https://doi.org/10.1002/wcms.1603.
34. Balaban, A. T. Topological Indices Based on Topological Distances in Molecular Graphs. *Pure Appl. Chem.* **1983**, *55* (2), 199–206. https://doi.org/10.1351/pac198855020199.
35. Balaban, A. T. Highly Discriminating Distance-Based Topological Index. *Chem. Phys. Lett.* **1982**, *89* (5), 399–404. https://doi.org/10.1016/0009-2614(82)80009-2.
36. Todeschini, R.; Gramatica, P. The Whim Theory: New 3D Molecular Descriptors for Qsar in Environmental Modelling. *SAR QSAR Environ. Res.* **1997**, *7* (1–4), 89–115. https://doi.org/10.1080/10629369708039126.
37. Grisoni, F.; Schneider, G. Molecular Scaffold Hopping via Holistic Molecular Representation. In *Protein-Ligand Interactions and Drug Design*; Ballante, F., Ed.; Methods in Molecular Biology; Springer US: New York, NY, 2021; pp 11–35. https://doi.org/10.1007/978-1-0716-1209-5_2.
38. Willighagen, E. L.; Mayfield, J. W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; Torrance, G.; Evelo, C. T.; Guha, R.; Steinbeck, C. The Chemistry Development Kit (CDK) v2.0: Atom Typing, Depiction, Molecular Formulas, and Substructure Searching. *J. Cheminformatics* **2017**, *9* (1), 33. https://doi.org/10.1186/s13321-017-0220-4.
39. Hinselmann, G.; Rosenbaum, L.; Jahn, A.; Fechner, N.; Zell, A. jCompoundMapper: An Open Source Java Library and Command-Line Tool for Chemical Fingerprints. *J. Cheminformatics* **2011**, *3* (1), 3. https://doi.org/10.1186/1758-2946-3-3.
40. Boldini, D.; Ballabio, D.; Consonni, V.; Todeschini, R.; Grisoni, F.; Sieber, S. Effectiveness of Molecular Fingerprints for Exploring the Chemical Space of Natural Products. ChemRxiv October 31, 2023. https://doi.org/10.26434/chemrxiv-2023-0m355.
41. Klekota, J.; Roth, F. P. Chemical Substructures That Enrich for Biological Activity. *Bioinformatics* **2008**, *24* (21), 2518–2525. https://doi.org/10.1093/bioinformatics/btn479.
42. Kim, S. Getting the Most out of PubChem for Virtual Screening. *Expert Opin. Drug Discov.* **2016**, *11* (9), 843–855. https://doi.org/10.1080/17460441.2016.1216967.
43. Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25* (2), 64–73. https://doi.org/10.1021/ci00046a002.
44. Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological Torsion: A New Molecular Descriptor for SAR Applications. Comparison with Other Descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27* (2), 82–85. https://doi.org/10.1021/ci00054a008.
45. Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **2005**, *18* (8), 1093–1110. https://doi.org/10.1016/j.neunet.2005.07.009.
46. Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754. https://doi.org/10.1021/ci100050t.
47. Reutlinger, M.; Koch, C. P.; Reker, D.; Todoroff, N.; Schneider, P.; Rodrigues, T.; Schneider, G. Chemically Advanced Template Search (CATS) for Scaffold-Hopping and Prospective Target Prediction for 'Orphan' Molecules. *Mol. Inform.* **2013**, *32* (2), 133–138. https://doi.org/10.1002/minf.201200141.
48. Mahé, P.; Ralaivola, L.; Stoven, V.; Vert, J.-P. The Pharmacophore Kernel for Virtual Screening with Support Vector Machines. *J. Chem. Inf. Model.* **2006**, *46* (5), 2003–2014. https://doi.org/10.1021/ci060138m.
49. Vidal, D.; Thormann, M.; Pons, M. LINGO, an Efficient Holographic Text Based Method To Calculate Biophysical Properties and Intermolecular Similarities. *J. Chem. Inf. Model.* **2005**, *45* (2), 386–393. https://doi.org/10.1021/ci0496797.
50. Probst, D.; Reymond, J.-L. A Probabilistic Molecular Fingerprint for Big Data Settings. *J. Cheminformatics* **2018**, *10* (1), 66. https://doi.org/10.1186/s13321-018-0321-8.

51. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why Do Tree-Based Models Still Outperform Deep Learning on Tabular Data? arXiv July 18, 2022. http://arxiv.org/abs/2207.08815 (accessed 2023-06-20).
52. Shwartz-Ziv, R.; Armon, A. Tabular Data: Deep Learning Is Not All You Need. *Inf. Fusion* **2022**, *81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011.
53. Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models. *J. Cheminformatics* **2021**, *13* (1), 12. https://doi.org/10.1186/s13321-020-00479-8.
54. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A Comparative Analysis of Gradient Boosting Algorithms. *Artif. Intell. Rev.* **2021**, *54* (3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5.
55. Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29* (5), 1189–1232. https://doi.org/10.1214/aos/1013203451.
56. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: San Francisco California USA, 2016; pp 785–794. https://doi.org/10.1145/2939672.2939785.
57. Biau, G.; Scornet, E. A Random Forest Guided Tour. *TEST* **2016**, *25* (2), 197–227. https://doi.org/10.1007/s11749-016-0481-7.
58. Klusowski, J. M. Analyzing CART. arXiv August 13, 2020. http://arxiv.org/abs/1906.10086 (accessed 2024-01-18).
59. Breiman, L. *Classification and Regression Trees*; Routledge: New York, 2017. https://doi.org/10.1201/9781315139470.
60. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. 11.
61. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.
62. Friedman, J. H. Stochastic Gradient Boosting. *Comput. Stat. Data Anal.* **2002**, *38* (4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2.
63. Breiman, L. Using Iterated Bagging to Debias Regressions. *Mach. Learn.* **2001**, *45* (3), 261–277. https://doi.org/10.1023/A:1017934522171.
64. Ustimenko, A.; Beliakov, A.; Prokhorenkova, L. Gradient Boosting Performs Gaussian Process Inference. arXiv October 13, 2022. https://doi.org/10.48550/arXiv.2206.05608.
65. Esposito, C.; Landrum, G. A.; Schneider, N.; Stiefl, N.; Riniker, S. GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning. *J. Chem. Inf. Model.* **2021**, *61* (6), 2623–2640. https://doi.org/10.1021/acs.jcim.1c00160.
66. Casanova-Alvarez, O.; Morales-Helguera, A.; Cabrera-Pérez, M. Á.; Molina-Ruiz, R.; Molina, C. A Novel Automated Framework for QSAR Modelling of Highly Imbalanced *Leishmania* High-Throughput Screening Data. *J. Chem. Inf. Model.* **2021**, *61* (7), 3213–3231. https://doi.org/10.1021/acs.jcim.0c01439.
67. Haibo He; Garcia, E. A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21* (9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239.
68. Idakwo, G.; Thangapandian, S.; Luttrell, J.; Li, Y.; Wang, N.; Zhou, Z.; Hong, H.; Yang, B.; Zhang, C.; Gong, P. Structure–Activity Relationship-Based Chemical Classification of Highly Imbalanced Tox21 Datasets. *J. Cheminformatics* **2020**, *12* (1), 66. https://doi.org/10.1186/s13321-020-00468-x.
69. Beckmann, M.; Ebecken, N. F. F.; Pires De Lima, B. S. L. A KNN Undersampling Approach for Data Balancing. *J. Intell. Learn. Syst. Appl.* **2015**, *07* (04), 104–116. https://doi.org/10.4236/jilsa.2015.74010.
70. Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. https://doi.org/10.1613/jair.953.

71. Haibo He; Yang Bai; Garcia, E. A.; Shutao Li. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*; IEEE: Hong Kong, China, 2008; pp 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969.
72. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *ArXiv170802002 Cs* **2018**.
73. Yeung, M.; Sala, E.; Schönlieb, C.-B.; Rundo, L. Unified Focal Loss: Generalising Dice and Cross Entropy-Based Losses to Handle Class Imbalanced Medical Image Segmentation. arXiv November 24, 2021. http://arxiv.org/abs/2102.04525 (accessed 2022-07-08).
74. Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; Kumar, S. Long-Tail Learning via Logit Adjustment. arXiv July 9, 2021. http://arxiv.org/abs/2007.07314 (accessed 2022-06-08).
75. Ghorbani, A.; Zou, J. Data Shapley: Equitable Valuation of Data for Machine Learning. 10.
76. Yoon, J.; Arık, S. Ö.; Pfister, T. Data Valuation Using Reinforcement Learning. 10.
77. Pruthi, G.; Liu, F.; Sundararajan, M.; Kale, S. Estimating Training Data Influence by Tracing Gradient Descent. arXiv November 14, 2020. http://arxiv.org/abs/2002.08484 (accessed 2022-08-31).
78. Akyurek, E.; Bolukbasi, T.; Liu, F.; Xiong, B.; Tenney, I.; Andreas, J.; Guu, K. Towards Tracing Knowledge in Language Models Back to the Training Data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*; Association for Computational Linguistics: Abu Dhabi, United Arab Emirates, 2022; pp 2429–2446.
79. Sink, R.; Gobec, S.; Pečar, S.; Zega, A. False Positives in the Early Stages of Drug Discovery. *Curr. Med. Chem.* **2010**, *17* (34), 4231–4255. https://doi.org/10.2174/092986710793348545.
80. Alves, V. M.; Capuzzi, S. J.; Braga, R. C.; Korn, D.; Hochuli, J. E.; Bowler, K. H.; Yasgar, A.; Rai, G.; Simeonov, A.; Muratov, E. N.; Zakharov, A. V.; Tropsha, A. SCAM Detective: Accurate Predictor of Small, Colloidally Aggregating Molecules. *J. Chem. Inf. Model.* **2020**, *60* (8), 4056–4063. https://doi.org/10.1021/acs.jcim.0c00415.
81. Borrel, A.; Mansouri, K.; Nolte, S.; Saddler, T.; Conway, M.; Schmitt, C.; Kleinstreuer, N. C. InterPred: A Webtool to Predict Chemical Autofluorescence and Luminescence Interference. *Nucleic Acids Res.* **2020**, *48* (W1), W586–W590. https://doi.org/10.1093/nar/gkaa378.
82. Stork, C.; Chen, Y.; Šícho, M.; Kirchmair, J. Hit Dexter 2.0: Machine-Learning Models for the Prediction of Frequent Hitters. *J. Chem. Inf. Model.* **2019**, *59* (3), 1030–1043. https://doi.org/10.1021/acs.jcim.8b00677.
83. Stork, C.; Wagner, J.; Friedrich, N.-O.; de Bruyn Kops, C.; Šícho, M.; Kirchmair, J. Hit Dexter: A Machine-Learning Model for the Prediction of Frequent Hitters. *ChemMedChem* **2018**, *13* (6), 564–571. https://doi.org/10.1002/cmdc.201700673.
84. Koh, P. W.; Liang, P. Understanding Black-Box Predictions via Influence Functions. arXiv December 29, 2020. http://arxiv.org/abs/1703.04730 (accessed 2022-06-20).
85. Karlaš, B.; Dao, D.; Interlandi, M.; Li, B.; Schelter, S.; Wu, W.; Zhang, C. Data Debugging with Shapley Importance over End-to-End Machine Learning Pipelines. arXiv April 26, 2022. http://arxiv.org/abs/2204.11131 (accessed 2022-06-21).
86. Shapley, L. S. 17. A Value for n-Person Games. In *Contributions to the Theory of Games (AM-28), Volume II*; Kuhn, H. W., Tucker, A. W., Eds.; Princeton University Press, 1953; pp 307–318. https://doi.org/10.1515/9781400881970-018.
87. Jia, R.; Dao, D.; Wang, B.; Hubis, F. A.; Gurel, N. M.; Li, B.; Zhang, C.; Spanos, C. J.; Song, D. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. arXiv March 29, 2020. https://doi.org/10.48550/arXiv.1908.08619.
88. Ghorbani, A.; Zou, J.; Esteva, A. Data Shapley Valuation for Efficient Batch Active Learning. arXiv April 16, 2021. http://arxiv.org/abs/2104.08312 (accessed 2024-01-18).

89. Jia, R.; Wu, F.; Sun, X.; Xu, J.; Dao, D.; Kailkhura, B.; Zhang, C.; Li, B.; Song, D. Scalability vs. Utility: Do We Have to Sacrifice One for the Other in Data Importance Quantification? In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Nashville, TN, USA, 2021; pp 8235–8243. https://doi.org/10.1109/CVPR46437.2021.00814.
90. Sharchilev, B.; Ustinovsky, Y.; Serdyukov, P.; de Rijke, M. Finding Influential Training Samples for Gradient Boosted Decision Trees. arXiv March 12, 2018. http://arxiv.org/abs/1802.06640 (accessed 2022-07-29).

PART II

PUBLICATIONS

3.

Tuning gradient boosting for imbalanced bioassay modelling with custom loss functions

Accepted open access article in *Journal of Cheminformatics* 14, 80 (**2022)**.

by **Davide Boldini**, Lukas Friedrich, Daniel Kuhn and Stephan A. Sieber.

https://doi.org/10.1186/s13321-022-00657-w

Synopsis

Although there is currently a wealth of HTS bioactivity data available, most of these datasets are extremely imbalanced towards inactive compounds, making the development of molecular property prediction models difficult. In this study, we sought to combine Gradient Boosting with bespoke loss functions, allowing the model to learn from imbalanced data effectively without compromising its applicability domain or increasing the training time.

To this end, we selected four variations of binary cross-entropy originally developed for neural network training and adapted them for training GBMs. Specifically, we considered Focal loss, Logit-Adjusted loss, Label Distribution Aware Margin loss and Equalization loss. The first downplays the impact well-classified samples have on the overall loss computation, the second and third derive a logit shift from the class distribution to apply to the cross-entropy calculation and the fourth dynamically adapts class weights according to the distribution of the gradients during training. To assess their effectiveness for molecular property prediction we considered weighted cross-entropy as the baseline, since it is the most common algorithmic modification for tackling class imbalance.

After evaluation on 42 molecular property prediction tasks and 2 million unique compounds, our proposed approach significantly outperformed the baseline on 5 out of 6 datasets. Crucially, thanks to this performance improvement we were able to push GBM to match or outperform the previous state-of-the-art QSAR models for these datasets, which included more computationally costly methods such as graph neural networks. Additionally, custom loss functions show much faster convergence rates than weighted cross-entropy, decreasing the training time of QSAR models up to a factor of 8. This is especially important for large, imbalanced bioactivity datasets such as HTS campaigns, making our approach uniquely suited to tackle these molecular property prediction tasks.

Taken together, the study demonstrates the effectiveness of combining custom loss function with Gradient Boosting for QSAR modelling, achieving state-of-the-art performance with minimal computational cost on a wide variety of molecular property prediction tasks. In the future, this approach could be used to leverage pre-existing bioactivity data for a wide variety of property prediction applications, e.g. for antibiotic or antivirulent activity virtual screening.

Author contributions

Davide Boldini conceived the project. Davide Boldini, Lukas Friedrich, Daniel Kuhn and Stephan A. Sieber defined the benchmarking protocols and chose the datasets for the study. Davide Boldini wrote the Python code for implementing the custom loss functions. Davide Boldini wrote the scripts for performing the benchmarks on the Tox21, HIV, MUV, Phos and

NTP datasets in terms of classification performance and training time. Lukas Friedrich evaluated the performance of the approach on the industry dataset using code provided by Davide Boldini. Finally, Davide Boldini wrote the first manuscript, which was further edited and discussed with input from all authors.

4.

Practical guidelines for the use of gradient boosting for molecular property prediction

Accepted open access article in *Journal of Cheminformatics* 15, 73 (**2023)**.

by **Davide Boldini**, Francesca Grisoni, Daniel Kuhn, Lukas Friedrich and Stephan A. Sieber.

https://doi.org/10.1186/s13321-023-00743-7

Synopsis

Gradient Boosting is a powerful machine learning algorithm, which has become the *de facto* standard option for modelling tabular data in a wide variety of computational fields and data science competitions. To date, there are three main variants of GBMs, namely XGBoost, LightGBM and CatBoost, each with its own unique algorithms and implementation details. The goal of this study was to critically evaluate and compare each variant for molecular property prediction, with the aim of elucidating the best practices for QSAR modelling with GBMs. To this end, we designed a large benchmarking study encompassing 16 datasets, 94 molecular property prediction tasks, 1.4 million unique compounds and trained 157,590 GBM models.

In terms of predictive performance, our analysis showed that XGBoost significantly outperformed both LightGBM and CatBoost on the majority of datasets by a 5 % margin. This performance increase was especially noticeable on larger datasets. However, this comes at the cost of computational efficiency, with this GBM variant being up to 100 times slower than the others on HTS data. LightGBM and CatBoost were generally equal in terms of predictive performance, but the former was much faster than the latter, outperforming it in terms of training time by a factor of 50 for large datasets.

In terms of explainability, our results surprisingly indicated that different GBM variants prioritized different chemical moieties for a given molecular property prediction task. This further highlights the algorithmic differences between these methods and reinforces the need of using expert knowledge to assess data-driven structural insights on bioactivity.

Finally, we analyzed the hyperparameter optimization process to determine the set of most important parameters to tune when using GBM for QSAR modelling. Our findings showed that while the optimal set of hyperparameters can change depending on featurization and property prediction task, tuning only the most important parameters lead to significantly improved performance while lowering the computational cost of hyperparameter optimization.

In conclusion, our work constituted the first large-scale comparison of GBM algorithms specifically for molecular property prediction and outlined the most important considerations to keep in mind when modelling QSAR data with this algorithm. Given the rising popularity of GBM for cheminformatics applications, this work provides an ideal starting point for its effective usage for molecular property prediction.

Author contributions

Davide Boldini conceived the project. Davide Boldini and Francesca Grisoni defined the benchmarking protocols and chose the datasets for the study. Davide Boldini wrote the scripts for performing the benchmarks on all datasets in terms of predictive performance, training time,

explainability and hyperparameter importance. Finally, Davide Boldini wrote the first manuscript, which was further edited and discussed with input from all authors.

5.

Machine learning assisted hit prioritization for high throughput screening in drug discovery

Accepted open access article in *ACS Central Science* 10, 4 (**2024)**.

by **Davide Boldini**, Lukas Friedrich, Daniel Kuhn and Stephan A. Sieber.

https://doi.org/10.1021/acscentsci.3c01517

Synopsis

False positives are abundant and heterogeneous in HTS campaigns, making the prioritization of true bioactive compounds a difficult endeavor. As such, *in silico* identification of interferents in these datasets is an important open challenge in cheminformatics. While several tools have been developed to help with this task, they tend to be inherently restricted to specific interference mechanisms, regions of the chemical space or assay technologies, thus limiting their effectiveness.

In this work, we developed a novel data valuation algorithm for the simultaneous identification of false positives and the prioritization of true bioactive compounds, named Minimal Variance Sampling Analysis (MVS-A). In short, our approach first fits a GBM classifier on the HTS data, then computes the importance each molecule has on the GBM predictor using MVS-A and finally uses these scores to distinguish between interferents and truly bioactive molecules. Thanks to its fully data-driven nature and its reliance on exclusively the HTS campaign of interest, MVS-A is not limited in terms of assay technology, compound class or false positive type, making it the first global tool for efficient prioritization of hits in HTS data.

To validate our algorithm, we curated a selection of 20 HTS campaigns and confirmatory screens, both from academia and the pharmaceutical industry, encompassing a diverse selection of compounds and covering a wide range of protein targets as well as assay types. On average, MVS-A significantly outperformed all other approaches both in terms of false positive detection and prioritization of true positives, confirming its versatility for processing HTS data. Additionally, our results showed that MVS-A identifies an extremely diverse selection of interferents in terms of chemical moieties and interference mechanisms. We then performed a retrospective study on a HTS campaign for the identification of presynaptic choline transporter (CHT) inhibitors, to validate the applicability of MVS-A for real case scenarios. Our analysis showed that our algorithm successfully prioritized the 6 most pharmacologically relevant inhibitors from the primary screening data from a library of more than 300.000 molecules. Finally, MVS-A is extremely computationally efficient, requiring less than 30 seconds to process entire HTS datasets on low-end hardware, further boosting its applicability in a wide range of screening campaigns.

Taken together, our findings showcased the usefulness of MVS-A for processing HTS data for hit prioritization and false positive identification. Future work could further investigate other data valuation approaches for cheminformatics applications, e.g. for active learning or for providing sample-driven model explanations.

Author contributions

Davide Boldini conceived the project. Davide Boldini and Lukas Friedrich curated the datasets and defined the benchmarking protocol. Davide Boldini wrote the Python code for implementing MVS-A, for training the false positive predictors, for executing the benchmarks and for performing the retrospective case study analysis. Lukas Friedrich evaluated the performance of the approach on the industry datasets using code provided by Davide Boldini. Finally, Davide Boldini wrote the first manuscript, which was further edited and discussed with input from all authors.

6.

Effectiveness of molecular fingerprints for exploring the chemical space of natural products

Accepted open access article in *Journal of Cheminformatics* 16, 35 (**2024**).

by **Davide Boldini**, Davide Ballabio, Viviana Consonni, Roberto Todeschini, Francesca Grisoni and Stephan A. Sieber.

https://doi.org/10.1186/s13321-024-00830-3

Synopsis

Natural products have garnered significant interest for biomedical applications, thanks to their high potency, chemical diversity and selectivity. However, they pose unique challenges for cheminformatics modelling, making the development of QSAR models and virtual screening for this class of compound difficult. This is because most modelling methods were designed with drug-like compounds in mind, while natural products have different chemical features, e.g. a higher fraction of sp^3-hybridized carbons and more stereocenters.

To tackle this gap, we benchmarked molecular fingerprints for virtual screening and bioactivity prediction on natural products. To ensure the comprehensiveness of our findings, we evaluated 20 different fingerprinting algorithms from 5 different categories, covering over 40 years of research. In terms of natural product diversity, we investigated over 100,000 unique compounds from the COlleCtion of Open Natural prodUcTs (COCONUT) and Comprehensive Marine Natural Product Database (CMNPD) repositories on 13 different bioactivity prediction tasks.

In terms of molecular similarity, our findings showed that there are relevant differences between and within fingerprint types when modelling natural products. For example, substructure-based algorithms are particularly heterogeneous and tend to have less informative bits, due to their focus on substructures typically found in drug-like compounds. Additionally, the pairwise similarity between fingerprints changed depending on the considered region of the chemical space. For example, MHFP and MAP4 fingerprints were substantially more correlated to Avalon fingerprints and less correlated with PubChem fingerprints when modelling natural products.

Regarding molecular property prediction, our results highlighted that while ECFPs are usually the standard encoding approach in cheminformatics, they can be matched or outperformed by other fingerprints when modelling natural products. While the performance of a given fingerprint depends on the chosen classification algorithm and bioactivity prediction task, ASP and MHFP were particularly promising for encoding natural products. Conversely, substructure-based approaches showed inferior performance, as a byproduct of their emphasis on drug-like moieties.

In conclusion, given the performance variability observed across different bioactivity prediction tasks, these findings reinforce the need of evaluating multiple fingerprint types when modelling natural products. Additionally, it is crucial to design molecular representations that accurately capture the unique chemical characteristics of this class of compounds.

Author contributions

Davide Boldini, Davide Ballabio, Viviana Consonni and Roberto Todeschini conceived the project. Davide Boldini, Francesca Grisoni, Davide Ballabio, Viviana Consonni and Roberto Todeschini defined the benchmarking protocols, while Davide Boldini and Francesca Grisoni chose the databases used for the study. Davide Boldini implemented the fingerprints in Python and wrote the scripts used for the similarity search comparison, unsupervised embedding calculation and bioactivity prediction benchmarking. Roberto Todeschini calculated the Minimum Spanning Tree for the correlation matrix between the Tanimoto similarity scores of different fingerprints. Finally, Davide Boldini and Davide Ballabio wrote the first draft of the manuscript, which was further edited and discussed with input from all authors.

7.

Research conclusion and outlook

While molecular property prediction has the potential of significantly speeding up the early stages of the drug discovery pipeline, its successful application for virtual screening hinges on efficiently leveraging the wealth of bioactivity data collected by HTS campaigns.[1–3] However, modelling these datasets is not straightforward due to their class imbalance, size, noisy readouts and lack of chemical diversity.[3–7] In this thesis, a broad range of computational methods are developed and investigated to tackle these four issues, focusing specifically on GBM as the modelling algorithm for HTS data.

In the first major project presented in this work, a new approach for tackling class imbalance for molecular property prediction was described (Chapter 3).[8] Our results showed that by modifying the optimization objective of GBM, it is possible to obtain robust QSAR models for imbalanced classification. Furthermore, using custom loss functions improved the performance of GBM to match or outperform other state-of-the-art algorithms, while shortening its training time. Finally, unlike resampling approaches, our method does not artificially increase the dataset size nor it decreases the applicability domain of the resulting QSAR model, making it an ideal option to handle class imbalance.

Chapter 4 investigated the best practices for using GBM for molecular property prediction.[9] To this end, the three main variants of the GBM algorithm, namely XGBoost, LightGBM and CatBoost, were benchmarked on a wide variety of molecular property prediction tasks. For large HTS datasets, XGBoost outperformed all alternatives by a 5% margin, while LightGBM was vastly more computationally efficient than the other GBM algorithms. Evaluation of the most important variables identified by these approaches showed that different GBM implementations highlight different chemical features, further underpinning the importance of relying on expert knowledge to avoid spurious explanations. Additionally, by analyzing the hyperparameter optimization process of GBMs on QSAR datasets, the most relevant parameters for modeling specific bioactivity endpoints with GBM were determined.

These findings were then used for the development of a new algorithm for evaluating sample importance for GBM and its application on the efficient prioritization of HTS hits for further pharmacological development (Chapter 5).[10] The efficacy of this approach was demonstrated on 20 HTS campaigns for both assay interferent and true hit identification, with the proposed algorithm outperforming rule-based and data-driven baselines. Additionally, the false positives detected by this method are both chemically diverse and arising from different interference mechanisms, further highlighting the advantage of the proposed approach. To further validate the efficacy of combining data valuation and GBM in the context of drug discovery, a retrospective study was performed on a HTS campaign to discover new potential Alzheimer's disease therapeutics. Crucially, the algorithm successfully prioritized all pharmacologically

relevant molecules from the primary HTS screen, thus confirming the applicability of the proposed approach in real case scenarios.

In Chapter 6, different molecular fingerprints were investigated for natural product modelling.[11] The most significant finding was that the ECFP algorithm, while being the most common encoding approach for drug-like molecules, can be matched or outperformed by other fingerprints when modelling natural products for bioactivity prediction. This result further reinforces the need to carefully consider multiple molecular fingerprints when developing robust QSAR models.

Taken together, the research projects described in this thesis provide a solid foundation for modelling HTS data with GBM for molecular property prediction. Using custom learning objectives is a novel and versatile approach for handling class imbalance in cheminformatics, which is uniquely suited for HTS modelling since it improves the computational efficiency of the algorithm (Chapter 3). The guidelines determined in Chapter 4 allow practitioners to select the most appropriate GBM variant given the size of the HTS dataset, understand its pitfalls and how to optimally tune its parameters. The novel data valuation approach described in Chapter 5 can then be used to remove problematic molecules in the dataset and to suggest compounds for further pharmacological development. Finally, the findings from Chapter 6 suggest molecular fingerprints to use when modelling HTS datasets when the target library for virtual screening heavily features natural products.

In terms of further research, the work described in this thesis offers several opportunities. For example, the optimization objective of GBM could be further improved by incorporating higher order terms of the polynomial expansion of the cross-entropy loss, as shown for deep learning methods.[12] Another promising research direction would be the use of zero-shot prediction models to refine the screening library before performing the HTS analysis, given the text description of the assay of interest.[13,14] This is currently being evaluated within the TwinBooster framework, which combines Large Language Model finetuning, self-supervised representation learning and GBM to achieve state-of-the-art zero-shot performance on QSAR datasets.[15] Finally, cheminformatics applications of data valuation algorithms are a particularly promising yet unexplored field of research. While this thesis focused specifically on employing it for identifying true positives and false positives in HTS datasets, there are many further uses for this class of algorithm. Potential applications include active learning for efficient sample acquisition during an HTS campaign and importance-driven undersampling. These research avenues are currently being investigated in an ongoing project, currently available as a preprint.[16]

To conclude, this thesis successfully investigated the use of GBMs for modelling HTS datasets in drug discovery. The findings and algorithms described in this work can be readily used for

accelerating early-stage drug discovery and provide an important foundation for further research into modelling strategies for this class of datasets.

7.1 References

1. Keshavarzi Arshadi, A.; Salem, M.; Firouzbakht, A.; Yuan, J. S. MolData, a Molecular Benchmark for Disease and Target Based Machine Learning. *J. Cheminformatics* **2022**, *14* (1), 10. https://doi.org/10.1186/s13321-022-00590-y.
2. Dreiman, G. H. S.; Bictash, M.; Fish, P. V.; Griffin, L.; Svensson, F. Changing the HTS Paradigm: AI-Driven Iterative Screening for Hit Finding. *Slas Discov.* **2021**, *26* (2), 257–262. https://doi.org/10.1177/2472555220949495.
3. Schneider, P.; Walters, W. P.; Plowright, A. T.; Sieroka, N.; Listgarten, J.; Goodnow, R. A.; Fisher, J.; Jansen, J. M.; Duca, J. S.; Rush, T. S.; Zentgraf, M.; Hill, J. E.; Krutoholow, E.; Kohler, M.; Blaney, J.; Funatsu, K.; Luebkemann, C.; Schneider, G. Rethinking Drug Design in the Artificial Intelligence Era. *Nat. Rev. Drug Discov.* **2020**, *19* (5), 353–364. https://doi.org/10.1038/s41573-019-0050-3.
4. Korkmaz, S. Deep Learning-Based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* **2020**, *60* (9), 4180–4190. https://doi.org/10.1021/acs.jcim.9b01162.
5. Sink, R.; Gobec, S.; Pečar, S.; Zega, A. False Positives in the Early Stages of Drug Discovery. *Curr. Med. Chem.* **2010**, *17* (34), 4231–4255. https://doi.org/10.2174/092986710793348545.
6. Casanova-Alvarez, O.; Morales-Helguera, A.; Cabrera-Pérez, M. Á.; Molina-Ruiz, R.; Molina, C. A Novel Automated Framework for QSAR Modelling of Highly Imbalanced *Leishmania* High-Throughput Screening Data. *J. Chem. Inf. Model.* **2021**, *61* (7), 3213–3231. https://doi.org/10.1021/acs.jcim.0c01439.
7. P. Wilson, B. A.; C. Thornburg, C.; J. Henrich, C.; Grkovic, T.; R. O'Keefe, B. Creating and Screening Natural Product Libraries. *Nat. Prod. Rep.* **2020**, *37* (7), 893–918. https://doi.org/10.1039/C9NP00068B.
8. Boldini, D.; Friedrich, L.; Kuhn, D.; Sieber, S. A. Tuning Gradient Boosting for Imbalanced Bioassay Modelling with Custom Loss Functions. *J. Cheminformatics* **2022**, *14* (1), 80. https://doi.org/10.1186/s13321-022-00657-w.
9. Boldini, D.; Grisoni, F.; Kuhn, D.; Friedrich, L.; Sieber, S. A. Practical Guidelines for the Use of Gradient Boosting for Molecular Property Prediction. *J. Cheminformatics* **2023**, *15* (1), 73. https://doi.org/10.1186/s13321-023-00743-7.
10. Boldini, D.; Friedrich, L.; Kuhn, D.; Sieber, S. *Machine Learning Assisted Hit Prioritization for High Throughput Screening in Drug Discovery*; preprint; Chemistry, 2023. https://doi.org/10.26434/chemrxiv-2023-zrgkp.
11. Boldini, D.; Ballabio, D.; Consonni, V.; Todeschini, R.; Grisoni, F.; Sieber, S. Effectiveness of Molecular Fingerprints for Exploring the Chemical Space of Natural Products. ChemRxiv October 31, 2023. https://doi.org/10.26434/chemrxiv-2023-0m355.
12. Leng, Z.; Tan, M.; Liu, C.; Cubuk, E. D.; Shi, X.; Cheng, S.; Anguelov, D. PolyLoss: A Polynomial Expansion Perspective of Classification Loss Functions. arXiv May 10, 2022. http://arxiv.org/abs/2204.12511 (accessed 2024-01-18).
13. Stanley, M.; Bronskill, J.; Maziarz, K.; Misztela, H.; Lanini, J.; Segler, M.; Schneider, N.; Brockschmidt, M. FS-Mol: A Few-Shot Learning Dataset of Molecules. 13.
14. Seidl, P.; Vall, A.; Hochreiter, S.; Klambauer, G. Enhancing Activity Prediction Models in Drug Discovery with the Ability to Understand Human Language. arXiv June 16, 2023. http://arxiv.org/abs/2303.03363 (accessed 2024-01-18).
15. Schuh, M. G.; Boldini, D.; Sieber, S. A. TwinBooster: Synergising Large Language Models with Barlow Twins and Gradient Boosting for Enhanced Molecular Property Prediction. arXiv January 9, 2024. http://arxiv.org/abs/2401.04478 (accessed 2024-01-18).
16. Hesse, J.; Boldini, D.; Sieber, S. *Data Valuation: A Novel Approach for Analyzing High Throughput Screen Data Using Machine Learning*; preprint; Chemistry, 2023. https://doi.org/10.26434/chemrxiv-2023-wlzlc.

Part III

APPENDIX

A.

Paper 1 (chapter 3)

Accepted open access article in *Journal of Cheminformatics* 14, 80 (**2022)**.

by **Davide Boldini**, Lukas Friedrich, Daniel Kuhn and Stephan A. Sieber.

https://doi.org/10.1186/s13321-022-00657-w

Journal of Cheminformatics

RESEARCH

Open Access

Tuning gradient boosting for imbalanced bioassay modelling with custom loss functions

Davide Boldini[1], Lukas Friedrich[2], Daniel Kuhn[2] and Stephan A. Sieber[1*]

Abstract

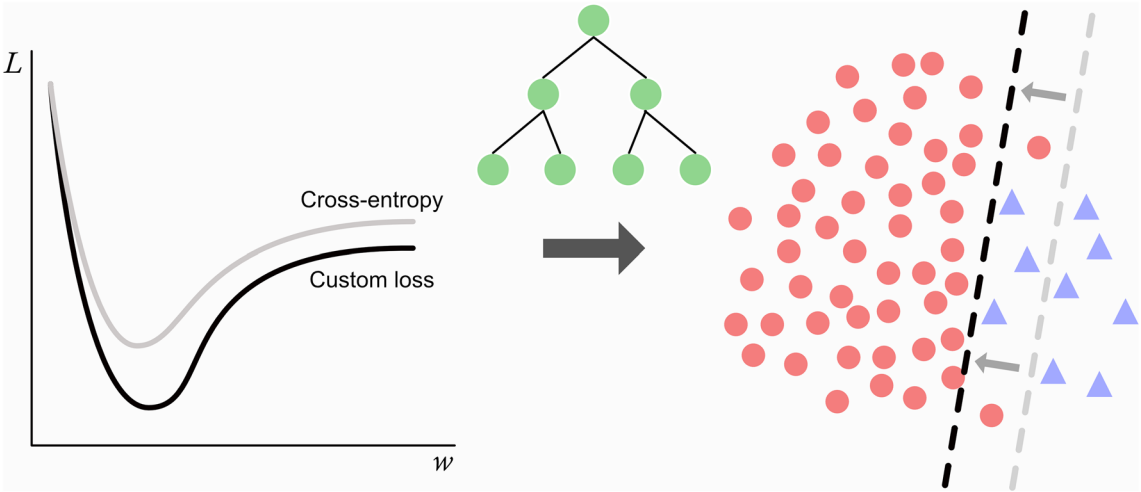
While in the last years there has been a dramatic increase in the number of available bioassay datasets, many of them suffer from extremely imbalanced distribution between active and inactive compounds. Thus, there is an urgent need for novel approaches to tackle class imbalance in drug discovery. Inspired by recent advances in computer vision, we investigated a panel of alternative loss functions for imbalanced classification in the context of Gradient Boosting and benchmarked them on six datasets from public and proprietary sources, for a total of 42 tasks and 2 million compounds. Our findings show that with these modifications, we achieve statistically significant improvements over the conventional cross-entropy loss function on five out of six datasets. Furthermore, by employing these bespoke loss functions we are able to push Gradient Boosting to match or outperform a wide variety of previously reported classifiers and neural networks. We also investigate the impact of changing the loss function on training time and find that it increases convergence speed up to 8 times faster. As such, these results show that tuning the loss function for Gradient Boosting is a straightforward and computationally efficient method to achieve state-of-the-art performance on imbalanced bioassay datasets without compromising on interpretability and scalability.

Keywords: Virtual screening, Imbalanced classification, Gradient boosting

*Correspondence: stephan.sieber@tum.de

[1] Center for Functional Protein Assemblies, Technical University of Munich (TUM), Ernst-Otto-Fischer-Straße 8, 85784 Garching, Germany
Full list of author information is available at the end of the article

Graphical Abstract



Introduction

In the last decade, machine learning (ML) and deep learning (DL) have radically transformed the conventional workflow for virtual screening in drug discovery [1]. This paradigm shift is strongly related to the substantial increase in freely available chemical data [2]. For example, popular repositories like PubChem and ZINC20 currently contain 1.2 million bioactivity assays and 1.4 billion unique compounds respectively [3–5]. Thanks to these resources, it is straightforward to obtain thousands of training points to develop high-performing predictive models, which can then be used to screen for novel ligands, antibiotics, antivirals and so forth [6–8].

The amount of data available has made it possible to use large neural networks, such as autoencoders (AE), transformers and graph neural networks (GNN) to learn data-driven molecular features, in contrast to prior featurization methods such as fingerprints and physicochemical descriptors [9–11]. Although these architectures have achieved impressive results on many benchmarks, they tend to be outperformed by descriptor-based models on class-imbalanced datasets [12–15], where the number of inactive compounds can be several orders of magnitude larger than the number of actives. Among the descriptor-based classifiers, tree ensembles such as Random Forest and, more recently, Gradient Boosting generally achieve the best performance [13, 15, 16]. Furthermore, this class of models provides additional benefits such as straightforward interpretability [17, 18], fewer hyperparameters to optimize and faster training speed compared to neural networks. [19]

The issue of class imbalance is of critical importance in drug discovery, given that the vast majority of the datasets available in this field are imbalanced [20], as highlighted by Landrum et al. [21] As such, there is an urgent need for novel strategies to tackle class imbalance for modelling bioassay data.

Current methods to address this issue usually rely on resampling the original class distribution or by employing algorithmic solutions such as custom loss functions [22, 23]. The latter approach has garnered interest in the field of computer vision, where the majority of classes in multitask classification have only a handful of positive samples [24–27]. Overall, these approaches rely on reframing the classification objective by reducing the influence of well-classified training instances, forcing the classifier to focus on hard-to-model samples, or by adjusting the unscaled output logits according to the prior probability to observe a given class. Research has shown that employing these methods provides a significant improvement over the baseline with virtually no additional computational cost. [24–27]

While there are several studies investigating resampling in the context of bioassay modelling [5, 28–30], changing the training objective has not been thoroughly investigated thus far. This study directly addresses this gap by investigating the effectiveness of a variety of recently published imbalance-insensitive loss functions for training Gradient Boosting classifiers. In this work, we considered Focal loss (FC) [24] Logit-adjusted loss (LA) [27] Equalization loss (EQ) [26] and Label-Distribution-Aware Margin (LDAM) [25] loss because of their

popularity in computer vision and their diversity from a theoretical standpoint.

The choice of pairing Gradient Boosting with the loss functions is motivated by its strong baseline performance across several studies in imbalanced classification tasks [13, 15]. Furthermore, its training speed makes [31, 32] it an attractive solution for modelling large-scale bioassays and its straightforward explicability allows detection of spurious correlations arising from false positives [33], which are known to be frequent in high-throughput screens [34, 35]. Therefore, tuning Gradient Boosting with bespoke loss functions can result in cheap, interpretable and high-performing models which is ideally suited for modelling imbalanced bioassay data.

We benchmark our proposed approach on six datasets from public (MoleculeNet [15] and MolData [20]) and proprietary (Merck KGaA) sources, comprising of approximatively 2 million compounds and 42 tasks with varying degrees of imbalance. Our findings show that changing the loss function provides a consistent, significant improvement, over cross entropy loss on five out of six datasets and that thanks to this modification, Gradient Boosting is able to match or outperform a wide variety of ML and DL approaches, including multitasking networks.

Methods

Gradient boosting

Originally developed by Friedman et al. [36] Gradient Boosting is a tree ensemble method that relies on training a sequence of weak learners (generally regression trees), each fitted on the residuals of the prior model. The final model is obtained by simply combining all the predictions from each individual classifier. Since this procedure is prone to overfitting, all Gradient Boosting frameworks offer a variety of regularization options, such as learning rates to modulate the influence of an individual learner on the final prediction, sampling of training samples and variables, L1 regularization and other options. [31, 32]

A key difference between Gradient Boosting and Random Forest is in the way individual trees are optimized. A Gradient Boosting classifier uses regression trees, where the individual splits are optimized according to the gradient and the Hessian of some loss function (i.e. cross-entropy), and converts the sum of predictions into a probability by applying the sigmoid function [31]. Random Forest instead uses decision trees, where the individual splits are optimized using criteria such as the Gini impurity or the Shannon entropy [37]. This distinction allows implementation of custom loss functions in a straightforward manner in any Gradient Boosting framework. [38]

There are several python packages available for training Gradient Boosting models, the most popular being XGBoost [31], CatBoost [39] and LightGBM [32]. In this study, we developed all models using the Python version of LightGBM 3.3.2.

Loss functions

The default loss function for many gradient-based classifiers, including LightGBM, when dealing with imbalaced classification is the weighted cross-entropy (WCE) [22, 23], which measures how close the class probabilities predicted by the classifier match the true class labels. It is defined as follows:

$$L_{CE} = -\sum_{n=1}^{m} w_i y_n \log(\widehat{y_n}) + (1 - y_n)\log(1 - \widehat{y_n}) \tag{1}$$

where m is the total number of samples, y_n are the target labels, $\widehat{y_n}$ are the predictions, w_i is a tunable parameter to account for class imbalance. When handling imbalanced datasets, classifiers tend to disregard the first term, corresponding to mistakes on the minority class, and only focus on minimizing the second term, corresponding to mistakes on the majority class, leading to a suboptimal model [22, 23]. This can be tackled by setting w_i equal to the ratio of inactive compounds versus active compounds.

Focal loss

Focal loss modifies the binary cross-entropy formulation by reducing the influence of well-classified samples on the overall loss [24, 38]. The formulation goes as follows:

$$L_F = -\sum_{n=1}^{m} y_n (1 - \widehat{y_n})^{\gamma} \log(\widehat{y_n}) + (1 - y_n)\widehat{y_n}^{\gamma} \log(1 - \widehat{y_n}) \tag{2}$$

where γ is a tunable parameter that affects the shape of the loss function. For high values of γ, the contribution of well classified samples to the overall loss approaches 0, allowing the gradient to focus more on the minority class. If γ is set to 0, the focal loss coincides with the standard cross-entropy loss.

Logit-adjusted loss

Instead of modulating sample influence during the training process like weighted cross-entropy or Focal loss, Logit-adjusted loss scales the raw logits from the classifier according to the a priori probabilities of the classes [27], as shown in Formula 3

$$L_{LA} = -\sum_{n=1}^{m} y_n \log(\sigma(p_n + \tau * \pi_m)) + (1 - y_n)\log(1 - \sigma(p_n + \tau * \pi_M))$$

where σ is the sigmoid function, p_i is the raw logit prediction, π_M and π_m are the prior probabilities for the majority and minority classes and τ is a smoothing factor that modulates the influence of the logit adjustments on the learning process. One key difference of Logit-adjusted loss compared to other approaches is that it guarantees Fisher consistency for the estimator by design, through a Bayes optimal solution for the balanced error. [27]

Label-distribution-aware margin loss

Similarly to Logit-adjusted loss, LDAM loss applies an offset to the raw logits from the model, but the optimal offsets are derived by minimizing a margin-based generalization bound [25]. One key limitation of margin-based approaches such as Support Vector Machines is that they rely on hinge loss [40], which is problematic to optimize for gradient-based methods because of its non-smoothness [25]. To tackle this issue, Cao et al. opted to use a cross-entropy inspired formulation, as shown in Formula 4:

$$L_{LDAM} = -\sum_{n=1}^{m} y_n \log\left(\sigma\left(p_n + \frac{C}{\sqrt[4]{n_m}}\right)\right) + (1-y_n)\log\left(1-\sigma\left(p_n + \frac{C}{\sqrt[4]{n_M}}\right)\right) \quad (4)$$

Where C is an hyperparameter to be tuned and n_m and n_M are the number of samples in the minority and majority class respectively.

Equalization loss

Another way to account for class imbalance is to operate at gradient level, for example by up-weighting gradients from minority samples and down-weighting the ones from majority samples according to the gradient ratio between classes. This approach has the theoretical advantage of weighting the minority class not only according to the class imbalance, but also according to the intrinsic difficulty of the classification problem, which might yield better weights compared to simple class counting statistics [26]. Another advantage is that this approach is function-agnostic, in the sense that it can be implemented to adjust any pre-existing loss function, i.e. cross-entropy.

To obtain the weighting coefficients for the gradients of the minority and majority classes, Equalization loss employs the following formula:

$$w_m{}^t = 1 + \alpha\left(1 - f\left(g_r{}^t\right)\right) \quad (5)$$

$$w_M{}^t = f\left(g_r{}^t\right) \quad (6)$$

where $g_r{}^t$ is the ratio of accumulated gradients between the minority and majority classes at iteration t, α is a hyperparameter that allows to increase the weight for the minority class and f is a mapping function:

$$f(x) = \frac{1}{1+e^{-\gamma(x-\mu)}} \quad (7)$$

With hyperparameters γ and μ.

To implement this approach, since Gradient Boosting is not trained with mini-batches, we considered the addition of one individual tree as one iteration, we clipped the gradients for numerical stability and we used binary cross-entropy as the underlying loss function.

Datasets

To evaluate our proposed approach, we collected six datasets from publicly available and proprietary sources. From MoleculeNet [15] we selected Tox21, HIV and MUV, from MolData [20] we chose Phosphatase and NTPase and finally we added one high-troughput screening (HTS) dataset from Merck KGaA, resulting in approximately 2 million compounds and 42 tasks. This selection covers a broad imbalance range and dataset size, to ensure that our findings are not biased by specific dataset conditions.

To access the publicly available data, we downloaded the cleaned MoleculeNet datasets from Jiang et al. [13] and the MolData ones from Arshadi and coworkers. [20]

The datasets are summarized in Table 1, reporting the average number of compounds and imbalance ratios across tasks. The individual values pertaining each endpoint can be found in Additional file 1: Table S1. Since the HTS benchmark is a proprietary dataset from Merck KGaA, the exact number of compounds is confidential.

Metrics

A critical step of developing classifiers for imbalanced classification is the choice of metric to measure performance [41, 42]. For example, evaluating machine learning models according to accuracy when dealing with

Table 1 Summary of the datasets employed in this study

Name	Source	Tasks	Compounds per task	Imbalance ratio
Tox21	MoleculeNet	12	6400	1:16
HIV	MoleculeNet	1	40748	1:27
MUV	MoleculeNet	17	14000	1:511
Phosphatase	MolData	5	330000	1:325
NTPase	MolData	6	330000	1:2963
HTS	Merck KGaA	1	> 330000	1:140

For a given dataset, the number of compounds per task and imbalance ratio are reported as averages across all tasks

imbalanced data can lead to misleading conclusions, since it does not properly account for the performance on the minority class [5, 41, 42]. To allow for comparisons against the results previously reported in the literature for these benchmarks, we opted to evaluate all datasets using all metrics used by Arshadi et al. [20] and Jiang and coworkers [13], with the addition of balanced accuracy, F1 score and the Matthews correlation coefficient (MCC). Therefore, for each benchmark receiver operating characteristic area under curve (ROC-AUC), precision-recall area under curve (PR-AUC), accuracy, balanced accuracy, recall, precision, F1 score and MCC were measured. A more in-depth discussion on the choice of metrics and their definition can be found in: Sect. 1 of the. Given the number Additional file 1 information of classifiers and metrics involved in our study, for conciseness we show in the main text only the metrics reported by the authors of the respective benchmarks. The performance tables with all metrics employed in this study can be found in: Sect. 3, 4 and 5 of the Additional file 1 information

Benchmarking procedure

After downloading the datasets from the respective repositories, all compounds were sanitized using RDKIT (version 2022.03.01) as described in the original papers and featurized using Extended-Connectivity Fingerprints (ECFP) with bit size 1024 and radius 2.

To develop the models, we followed two different benchmarking procedures depending on the dataset source. This way, the results obtained in this study are directly comparable to the performance of other classifiers reported in the respective papers. This enables us to put in perspective the improvements our approach provides over the default LightGBM implementation in a more conventional classifier comparison study.

For Tox21, HIV and MUV, we optimized each classifier in cross-validation using random splits, with a ratio of 80:10:10 for the training, validation and test set. Each model used early stopping on the loss of the validation set, while the test set was used to evaluate the performance of the model. To optimize the models we used Hyperopt (version 0.2.7) [43] for 20 iterations. Once the optimization was finished, we ran the model with optimal hyperparameters on 50 random splits, with a ratio of 80:10:10 for the training, validation and test set. Similar to the optimization phase, we used the validation set for early stopping and the test set for performance assessment. Regarding the choice of metrics, when comparing our approach to results from the literature we followed the guidelines from Wu et al. [15]: Tox21 and HIV were evaluated according to ROC-AUC, while MUV with PR-AUC.

For the Phosphatase and NTPase datasets, we employed the scaffold splits provided by Arshadi et al. [20] For each task, we optimized each model on the validation set and reported the performance on the test set. In all instances we used early stopping on the validation set to determine the optimal number of trees. All classifiers were optimized using Hyperopt [43] for 20 iterations and then evaluated 5 times using different random seeds. For comparisons with other machine learning algorithms, we reported the metrics employed by Arshadi et al. (accuracy, ROC-AUC, precision, recall) with the addition of the F1 score, to estimate the tradeoff between high precision and high recall.

For the Merck KGaA HTS dataset we employed the evaluation procedure for the MolData benchmarks. We created training, validation and testing sets using scaffold splitting with an 80:10:10 ratio. Then, we optimized all classifiers with Hyperopt for 20 iterations on the validation set using early stopping. Finally, we retrained each model with optimal parameters 5 times and measured all metrics on the test set.

To assess the efficacy of the custom loss functions, we use as baseline in all our benchmarks the performance of weighted cross-entropy and we evaluate whether the improvement is significant with 1-tailed Welch *t*-tests with Bonferroni correction. Furthermore, to contextualize the performance of LightGBM with custom loss functions, we compare the best performing model from our study to the models reported by Jiang et al. for MoleculeNet and by Arshadi et al. for MolData. All models from these papers employed weighted cross-entropy or class balancing schemes to model activity imbalance, depending on the underlying classification algorithm.

In the first study, four descriptor-based machine-learning methods and four graph-based neural networks were investigated. The descriptor-based models were Random Forest (RF), Support Vector Machine (SVM), XGBoost (XGB) and a neural network with dense layers (DNN), using a combination of 1D and 2D descriptors as well as two sets of fingerprints [13]. For the graph-based models, they considered a graph convolutional network (GCN), a graph attention network (GAT), a message-passing neural network (MPNN) and attentive fingerprints (AFP) [13]. For conciseness, for each MoleculeNet dataset we report the performance of the best descriptor-based model and graph-based model and we compare them to the best-performing LightGBM model using 2-tailed Welch *t*-tests with Bonferroni correction.

In the second study, the authors developed a multitask DNN on ECFP fingerprints with bit size 1024 and radius 2 and a multitask GCN. For these baselines, we omit statistical tests since the authors did not report standard deviations for their results.

Table 2 Summary of the benchmarking procedure for each dataset employed in this study

Name	Split	Replicates	Metrics for external comparison	External baselines
HIV	Random	50	ROC-AUC	RF, SVM, XGB, DNN, GCN, GAT, MPNN, AFP
Tox21	Random	50	ROC-AUC	RF, SVM, XGB, DNN, GCN, GAT, MPNN, AFP
MUV	Random	50	PR-AUC	RF, SVM, XGB, DNN, GCN, GAT, MPNN, AFP
Phosphatase	Scaffold	5	Accuracy, precision, recall, F1 score, ROC-AUC	DNN, GCN
NTPase	Scaffold	5	Accuracy, precision, recall, F1 score, ROC-AUC	DNN, GCN
HTS	Scaffold	5	Not applicable	Not applicable

Table 3 Summary of the results for the datasets belonging to the MoleculeNet repository

Name	Metric	WCE	FC	LA	EQ	LDAM	Best descriptor-based	Best graph-based
HIV	ROC-AUC	0.811 ± 0.02	0.831 ± 0.01	0.823 ± 0.03	0.809 ± 0.02	**0.833 ± 0.02**	0.822 ± 0.02	**0.833 ± 0.02**
Tox21	ROC-AUC	0.790 ± 0.01	0.808 ± 0.01	0.812 ± 0.01	0.781 ± 0.02	0.808 ± 0.01	0.838 ± 0.01	**0.852 ± 0.01**
MUV	PR-AUC	**0.152 ± 0.03**	0.127 ± 0.02	0.140 ± 0.03	0.126 ± 0.03	0.141 ± 0.03	0.112 ± 0.04	0.061 ± 0.03

The best values for each metric in each dataset are highlighted in bold

The benchmarking details for all datasets are summarized in Table 2.

Results

Moleculenet benchmarks

The results for the datasets from MoleculeNet are summarized in Table 3 and Fig. 1, while the *p*-values for the statistical tests are outlined in Additional file 1: Tables S8, S9, S10 and S14. The performance across all metrics for these datasets is shown in Additional file 1: Tables S2, S3 and S4.

Focal loss, Logit-adjusted loss and LDAM loss significantly outperform the weighted cross-entropy baseline for the HIV dataset. The best performing loss function is LDAM loss (0.833 ROC AUC), closely followed by Focal loss. Equalization loss achieves the lowest ROC-AUC out of all custom loss functions. Considering all metrics, Focal loss achieves the best performance in terms of PR-AUC, accuracy, F1 score and MCC and Equalization loss achieves the best precision value. With the exception of the F1 score, all differences are statistically significant. In terms of recall and balanced accuracy however, WCE outperforms all alternatives. Compared to the best descriptor-based model (SVM) and graph-based model (GCN) from Jiang et al., the LightGBM model with LDAM loss significantly outperforms the former and matches the ROC-AUC from the latter. The improvement on this dataset is especially significant, given that the weighted cross-entropy baseline is outperformed by both alternatives from Jiang et al.

For Tox21, similarly to the previous dataset, all custom losses with the exception of Equalization loss significantly outperform the weighted cross-entropy baseline in terms of ROC-AUC. Logit-adjusted loss achieves the best ROC-AUC with 0.812, narrowly outperforming LDAM loss and Focal loss. In terms of global performance however, LDAM loss has the most success, outperforming all alternatives on four metrics (PR-AUC, accuracy, precision, MCC), but except for precision and accuracy the differences are not statistically significant compared to the baseline. WCE achieves the best performance in terms of balanced accuracy, recall and F1 score. When comparing to the best models from Jiang et al., both options (RF and AFP) significantly outperform the Gradient Boosting

(See figure on next page.)

Fig. 1 Summary of the benchmarking results for the MoleculeNet datasets. Error bars represent the standard error of the mean (N = 50), while the asterisks denote whether the difference is significant (one indicates $\alpha < 0.05$, two $\alpha < 0.01$). The statistical tests with Bonferroni correction are carried out with respect to WCE or to the best performing loss function. We define the differences between loss functions within LightGBM as performance comparisons, while classifier comparisons refer to the benchmarking of the best loss function against the classifiers from Jiang et al. **a** Loss function comparison on the HIV dataset. **b** Comparison between the best loss function and the best models from Jiang et al. on the HIV dataset **c** Loss function comparison on the Tox21 dataset. **d** Comparison between the best loss function and the best models from Jiang et al. on the Tox21 dataset. **e** Loss function comparison on the MUV dataset. **f** Comparison between the best loss function and the best models from Jiang et al. on the MUV dataset

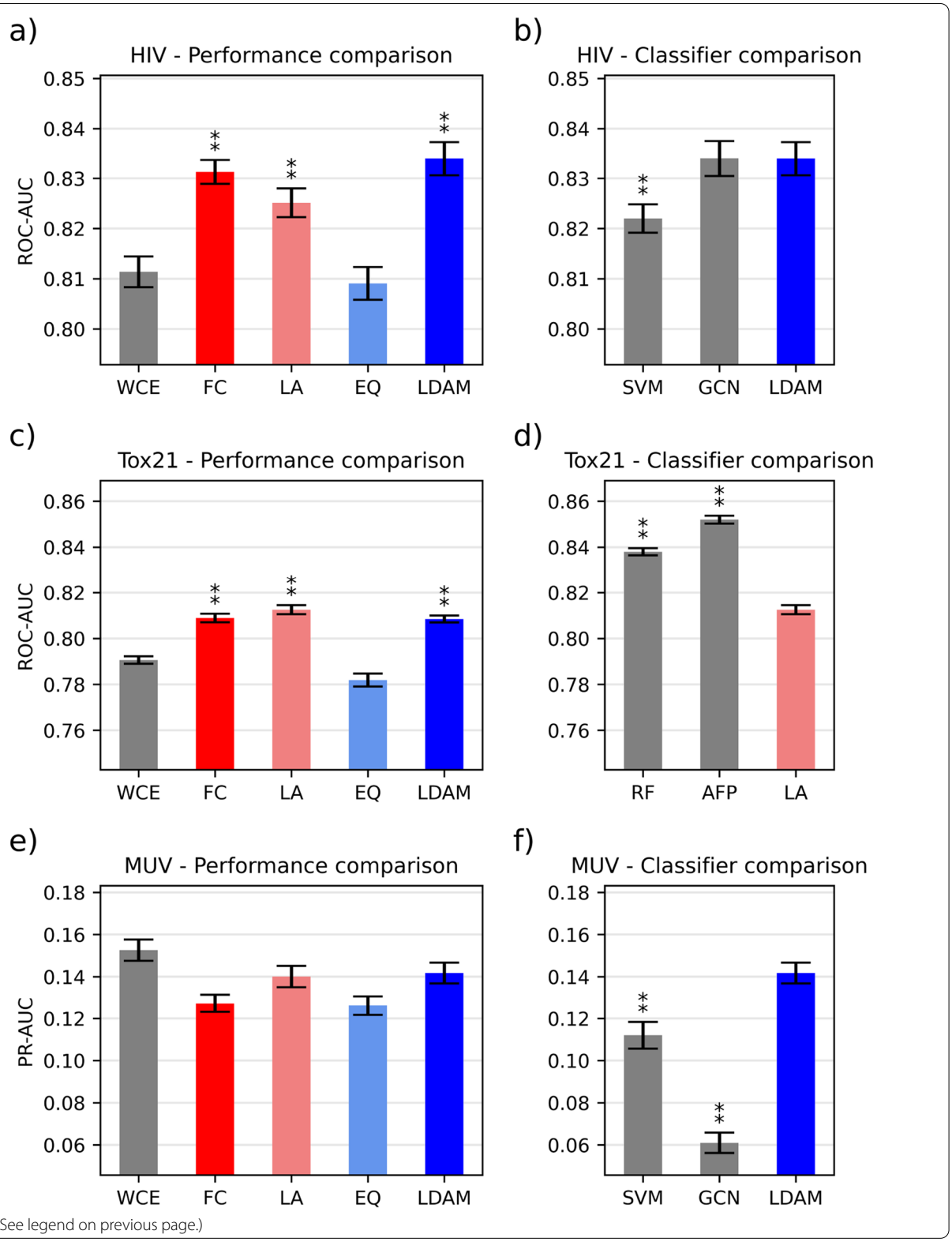


Fig. 1 (See legend on previous page.)

classifier with Logit-adjusted loss, possibly pointing to the fact that LightGBM might not be a good option for this dataset. Unlike XGBoost, LightGBM employs a leaf-wise tree splitting procedure, which is known to potentially lead to more complex structures that might overfit on small datasets [31, 32]. Among the datasets tested, Tox21 has the least compounds per task, which might explain why LightGBM performs comparatively poorly.

Regarding MUV, none of the custom losses are able to outperform the weighted cross-entropy baseline in any metric except accuracy. This is especially surprising considering that MUV is the most imbalanced dataset considered in this study, where one would expect to observe the greatest improvement over the baseline. This could be explained by the fact that the custom loss functions must optimize additional hyperparameters related to the loss, which have a strong impact on the performance of the classifier [27]. Since all classifiers generally achieve low PR-AUC values for this dataset, tuning these additional parameters could lead to a very noisy optimization process leading to an inferior optimum for a given number of iterations. Increasing the number of optimization evaluations could mitigate this issue.

Among the custom loss functions, LDAM loss performs the best with a PR-AUC value of 0.141, closely followed by Logit-adjusted loss. Interestingly, all LightGBM models are able to outperform all models from Jiang et al. Indeed, for this dataset LightGBM achieves more than double the performance reported for XGBoost in their paper. This again could be related to the differences in the tree-splitting procedure between the two implementations. Finally, the dataset also highlights the issues of data-driven representations when dealing with extreme imbalance, since in this benchmark all graph-based approaches achieve substantially lower performance than descriptor-based classifiers.

Moldata benchmarks

The custom loss functions were next evaluated using the MolData datasets.

All custom loss functions significantly outperform the weighted cross-entropy baseline for the Phosphatase dataset in terms of accuracy, precision (except Logit-adjusted loss) and ROC-AUC (Table 4, Additional file 1: Table S5 and Fig. 2, *p*-values for the statistical tests outlined in Additional file 1: Table S11). The only metrics where the baseline still outperforms the alternatives are recall and balanced accuracy. The F1 score for Logit-adjusted loss is higher, indicating that the trade-off between precision and recall is generally favorable, however the difference is not statistically significant. In terms of MCC and PR-AUC, LA loss achieves the best performance, significantly outperforming the baseline on both metrics. Compared to the multitask networks from Arshadi and coworkers, Focal loss outperforms them in all metrics except recall. The improvement is especially noticeable in terms of precision, achieving more than double the value reported for the GCN model.

For the NTPase benchmark, Logit-adjusted loss stands out as the best option, significantly outperforming the baseline in terms of precision, ROC-AUC and MCC (Table 4, Additional file 1: Table S6 and Fig. 2, p-values in Additional file 1: Table S12). LDAM loss and Focal loss also improve over the baseline, but the trend is not as consistent as for Logit-adjusted loss across all metrics. When comparing it to the baselines from Arshadi and coworkers, similarly to the results for the Phosphatase dataset, Logit-adjusted loss outperforms both multitask networks in all metrics except recall. The improvement is especially noticeable for ROC-AUC, going from 0.76 to 0.85.

Table 4 Summary of the benchmarking results for the datasets in the MolData repository

Name	Metric	WCE	FC	LA	EQ	LDAM	DNN—Arshadi	GCN –Arshadi
Phosphatase	Accuracy	0.989 ± 0.0005	**0.992 ± 4E-4**	**0.992 ± 3E-4**	**0.992 ± 7E-4**	**0.992 ± 2E-4**	0.885	0.984
	Precision	0.356 ± 0.01	0.455 ± 0.05	0.431 ± 0.06	**0.571 ± 0.01**	0.567 ± 0.05	0.027	0.144
	Recall	0.139 ± 0.006	0.125 ± 0.01	0.135 ± 0.01	0.085 ± 0.02	0.109 ± 0.03	**0.459**	0.191
	F1 score	0.200 ± 0.003	0.196 ± 0.01	**0.206 ± 0.01**	0.148 ± 0.01	0.182 ± 0.02	0.052	0.164
	ROC-AUC	0.814 ± 0.0005	**0.830 ± 0.001**	**0.830 ± 0.01**	0.821 ± 0.0003	0.825 ± 0.0008	0.739	0.815
NTPase	Accuracy	0.945 ± 0.001	0.945 ± 0.004	0.945 ± 0.0004	0.899 ± 0.02	**0.946 ± 0.005**	0.854	0.933
	Precision	0.381 ± 0.01	0.417 ± 0.01	0.472 ± 0.01	0.344 ± 0.04	**0.488 ± 0.006**	0.138	0.267
	Recall	0.300 ± 0.007	0.294 ± 0.005	0.267 ± 0.003	0.250 ± 0.02	0.255 ± 0.005	**0.526**	0.095
	F1 score	0.336 ± 0.003	**0.345 ± 0.004**	0.341 ± 0.005	0.289 ± 0.03	0.335 ± 0.003	0.219	0.141
	ROC-AUC	0.821 ± 0.01	0.787 ± 0.01	**0.852 ± 0.01**	0.764 ± 0.007	0.827 ± 0.02	0.763	0.763

The best values for each metric in each dataset are highlighted in bold

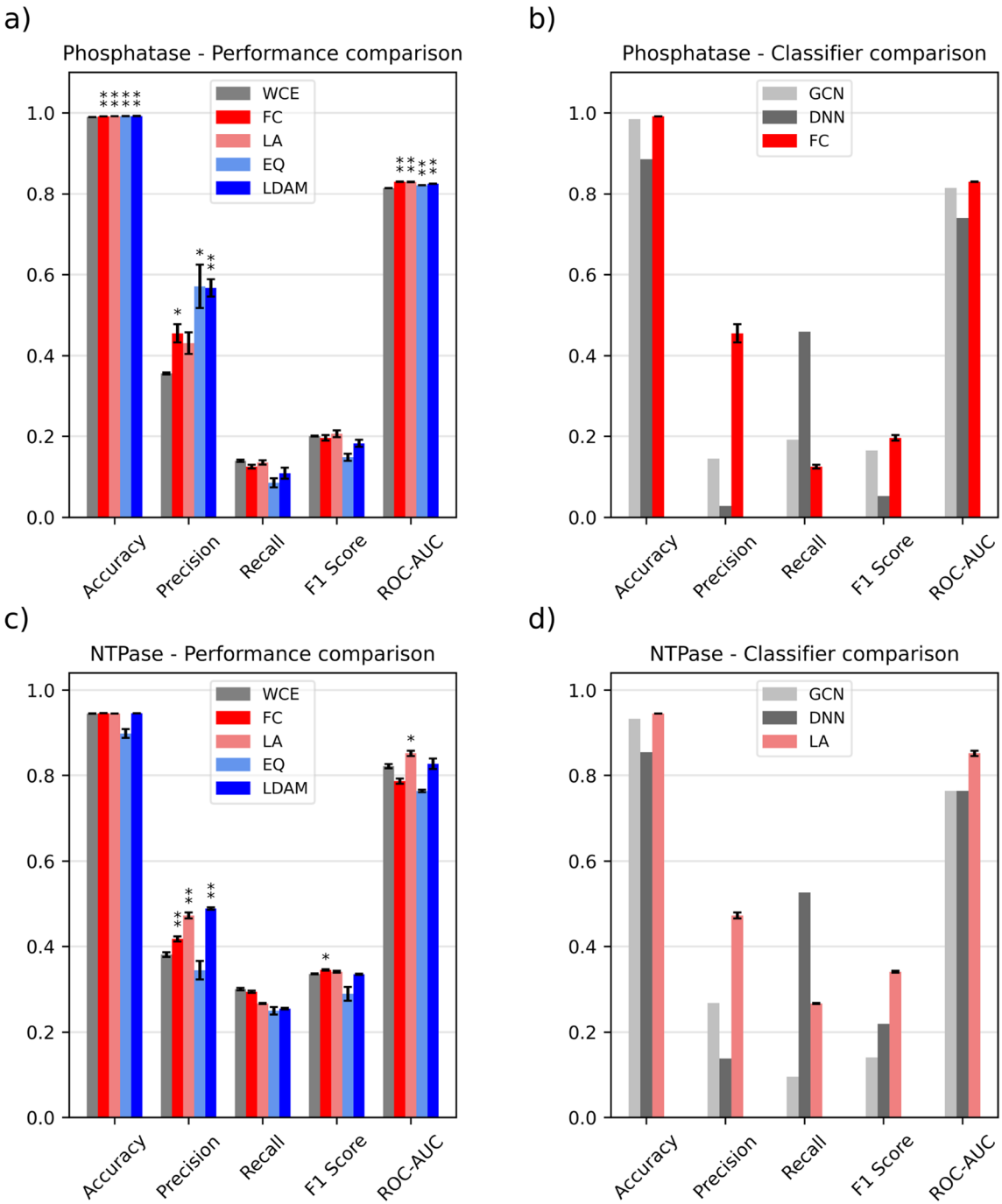


Fig. 2 Summary of the benchmarking results for the MolData datasets. Error bars represent the standard error of the mean (N = 5), while the asterisks denote whether the difference is significant (one indicates $\alpha < 0.05$, two $\alpha < 0.01$). The statistical tests with Bonferroni correction are carried out with respect to WCE. We define the differences between loss functions within LightGBM as performance comparisons, while classifier comparisons refer to the benchmarking of the best loss function against the classifiers from Arshadi et al. **a** Loss function comparison on the Phosphatase dataset. **b** Comparison between the best loss function and the best models from Arshadi et al. on the Phosphatase dataset **c** Loss function comparison on the NTPase dataset. **d** Comparison between the best loss function and the best models from Arshadi et al. on the NTPase dataset

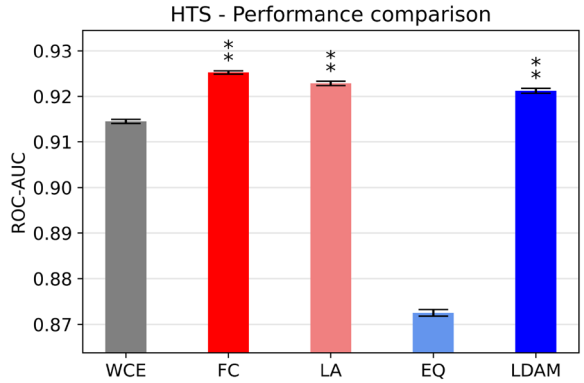


Fig. 3 Benchmarking results for the proprietary HTS dataset. Error bars represent the standard error of the mean (N = 5), while the asterisks denote whether the difference is significant (one indicates $\alpha < 0.05$, two $\alpha < 0.01$). The statistical tests are carried out with respect to WCE

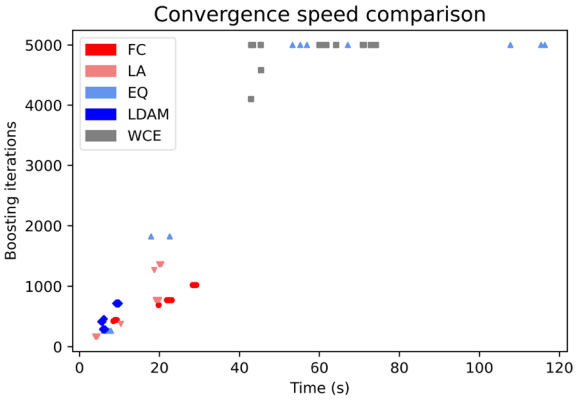


Fig. 4 Convergence speed comparison between weighted cross-entropy and the custom loss functions. Each dot represents a fit iteration in terms of boosting iterations required to trigger early stopping and computational time

Proprietary dataset benchmark

All loss functions, except Equalization loss, achieve excellent performance on the real-world industrial dataset, with ROC-AUC values above 0.9 (Fig. 3 and Additional file 1: Table S14, p-values for the statistical tests can be found in Additional file 1: Table S15). Focal loss, LDAM loss and Logit-adjusted loss significantly outperform the weighted cross-entropy baseline, consistently with the trends observed in the academic datasets. However, the relative increases between the baseline and the custom loss functions are minimal in terms of magnitude. This is likely because these classifiers already achieve near perfect performance, making it difficult to achieve substantial improvements. Considering the other metrics, Focal loss achieves the best performance on all metrics except balanced accuracy and recall, significantly outperforming the baseline in PR-AUC, precision, F1 score, MCC and accuracy. Logit-adjusted loss performs similarly to Focal loss, matching its performance in terms of MCC and PR-AUC while obtaining higher balanced accuracy.

Influence on convergence speed

To assess whether changing the loss function affects the number of boosting iterations required for convergence, we analyzed the number of trees and time required to fit the HIV dataset for each loss function. To do so, we optimized the hyperparameters of each classifier and measured the training time and number of trees on five 80:20 training-validation splits, using the external set for early stopping. The whole procedure was repeated three times, to ensure that the findings are independent of specific optima obtained during the optimization phase, for a total of 15 measurements per loss function. The results are summarized in Fig. 4, Additional file 1: Table S16 and Additional file 1: Table S17. Interestingly, the weighted cross-entropy baseline is the most computationally expensive option on average, requiring on average around 4900 boosting iterations and 59 s to fit the dataset. LDAM loss is the fastest loss function on average (7 s), closely followed by Logit-adjusted loss (13 s) and Focal loss (19 s). Equalization loss has the widest spread in terms of boosting iterations and training time, likely arising from training instability for this loss function.

Discussion

Remarkably, on five out of six datasets investigated, at least one custom loss function outperformed the weighted cross-entropy baseline. These findings display that our approach is robust to a wide variety of endpoints, dataset sizes and imbalance rates, including real world data. On average, the Equalization loss performed the worst, while Logit-adjusted loss achieved consistently strong performance across all datasets, followed by LDAM loss and Focal loss.

One possible explanation for the lower effectiveness of Equalization loss might be that approximating one mini-batch with the fitting of one boosted tree is not appropriate, thus rendering the accumulated gradient ratios unreliable. This is further confirmed by the high instability of the gradients we observed while implementing this loss for Gradient Boosting, which we attempted to correct using gradient clipping. Moreover, it is interesting that this custom loss function, which is the most similar to weighted cross-entropy since it relies on dynamically weighting the two class contributions, is also the one achieving the poorest performance. This further highlights the need for alternative approaches such as applying a class-specific offset to the raw logits (LDAM loss

and Logit-adjusted loss), or dampening the influence of well-classified samples (Focal loss).

When analyzing our results across all metrics, one relevant finding is that using custom loss functions leads to an overall increase in precision at the expense of recall when comparing to the weighted cross-entropy baseline. Depending on the context and purpose for which these datasets are modeled, the increase in precision might be extremely beneficial, i.e. in settings where experimental testing is expensive so it is paramount to reduce the number of false positives. Another interesting trend is the systematic increase in accuracy compared to the baseline, however this is not significant considering the inadequacy of this metric for imbalanced classification. In terms of global performance however, our proposed modifications still lead to better models overall, as indicated by generally higher MCC, ROC-AUC, PR-AUC and F1 scores across five out of six datasets. Furthermore, the increase in performance in terms of MCC is especially significant, given that this metric is known to perform extremely well in ranking classifiers when dealing with class imbalance [41]. It should be noted however that if the target metric is balanced accuracy, the baseline would be a more indicated choice of loss function since it consistently outperforms all alternatives.

Regarding the comparison with the external baselines from Arshadi et al. and Jiang et al., implementing the custom loss functions discussed in this study allows LightGBM to match or outperform the best models from those studies in four out of five datasets. This result is noteworthy considering the wide variety and complexity of the approaches employed by Jiang et al. and the fact that Gradient Boosting does not benefit from multitask learning, unlike the approaches from Arshadi et al. These findings highlight the importance of properly addressing imbalance with bespoke approaches rather than relying on simpler loss weighting schemes.

Regarding the convergence time, all losses required less iterations and training time than the weighted cross-entropy baseline, speeding up the computation by a factor of 8 for LDAM loss, 4 for Logit-adjusted loss, 3 for Focal loss and 1.2 for Equalization loss. One possible explanation for this could be that the modifications of cross-entropy investigated in this study provide more informative gradients, leading to faster convergence [44, 45]. This phenomenon could be caused by the inclusion of prior class probabilities in the loss formulation (Logit-adjusted and LDAM losses), or by forcing the total loss to be more dependent on hard to classify examples (Focal loss).

In summary, considering both the performance improvement and the influence on convergence time, Logit-adjusted and LDAM loss are the best options for tuning Gradient Boosting for imbalanced bioassay modelling. Interestingly, both approaches rely on logit shifting, which seems to indicate that this strategy is preferable than weighting approaches like Equalization loss or Focal loss, in agreement with the findings from Menon and coworkers [27]. Furthermore, both options, given sufficient hyperparameter optimization, can converge back to the original cross-entropy formulation, meaning that they are a suitable option even on datasets where the baseline might achieve better performance.

Finally, LightGBM with these modifications is a strong, efficient and interpretable baseline for future works on ligand-based virtual screening. This will provide an out-of-the-box solution for quickly modelling large bioassay data and will serve as a meaningful benchmark for more complex algorithms on imbalanced datasets.

Conclusion

In this study, we investigated the effectiveness of custom loss functions applied to Gradient Boosting for modelling extremely imbalanced bioassay data. To answer this question, we evaluated our approach against weighted cross-entropy, the current de-facto standard for imbalanced data classification, and a variety of classifiers from previous studies involving approximately 2 million compounds and 42 tasks from public and proprietary sources.

Our results show that all bespoke loss functions achieve statistically significant improvement over weighted cross-entropy across 5 out of 6 benchmarks, the most promising being Logit-adjusted loss and LDAM loss. Furthermore, thanks to these modifications, Gradient Boosting is able to match or outperform the best classifiers of other benchmarks for four out of five datasets. Additionally, the use of custom loss reduces the training time and computational cost for gradient boosting, as highlighted in our convergence iteration comparison.

The significance of these results is three-fold. First, they show the importance of appropriately tackling class imbalance with custom loss functions, an approach that has not been thoroughly investigated in the context of drug discovery until now. These modifications are particularly promising considering their widespread success in computer vision and could substitute or complement resampling-based approaches, which are already well established for bioassay modelling [5, 29, 30]. Second, they highlight the efficacy of Gradient Boosting coupled with proper loss functions for modelling extremely imbalanced bioassay data. This is relevant because Gradient Boosting has a unique set of advantages over other classifiers such as excellent scalability to large datasets [31, 32, 39], straightforward interpretability [17] and ease of optimization [19]. Third, our analysis shows that logit-shifting modifications of the cross-entropy loss

are generally more performant than weighting-based approaches for gradient boosting. This provides a solid foundation for developing novel loss functions and simplifies the choice of loss function when modelling imbalanced data.

Finally, our implementation, available at https://github.com/dahvida/gradient_boosting_CLF, is designed to handle any function definition with minimal external package dependencies to streamline the implementation of alternative loss functions for Gradient Boosting. We hope this will accelerate further research on newer loss functions for class imbalance, i.e. combo losses [46], as well as for regular classification, for example 0–1 losses with Langevin gradient descent [47].

Abbreviations

ML: Machine learning; DL: Deep learning; AE: Autoencoder; GNN: Graph neural network; FC: Focal loss; LA: Logit-adjusted loss; EQ: Equalization loss; LDAM: Label-distribution aware margin loss; WCE: Weighted cross-entropy; ECFP: Extended connectivity fingerprint; RF: Random forest; SVM: Support Vector machine; XGB: XGBoost; DNN: Dense neural network; GCN: Graph convolutional neural network; GAT: Graph attention neural network; MPNN: Message-passing neural network; AFP: Attentive fingerprint; ROC-AUC: Receiver operating characteristic area under curve; PR-AUC: Precision–recall area under curve.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-022-00657-w.

Additional file 1: Table S1. Description of the number of compounds and imbalance ratio, defined as the number of inactive compounds divided by the number of active ones, for each endpoint in each dataset. **Table S2.** Summary of the benchmarking results for the HIV dataset. The best values for each metric in each dataset are highlighted in bold. **Table S3.** Summary of the benchmarking results for the Tox21 dataset. The best values for each metric in each dataset are highlighted in bold. **Table S4.** Summary of the benchmarking results for the MUV dataset. The best values for each metric in each dataset are highlighted in bold. **Table S5.** Summary of the benchmarking results for the Phosphatase dataset. The best values for each metric in each dataset are highlighted in bold. **Table S6.** Summary of the benchmarking results for the NTPase dataset. The best values for each metric in each dataset are highlighted in bold. **Table S7.** Significance levels for the Welch tests after Bonferroni correction for each dataset. **Table S8.** P-values of the Welch tests (N = 50) for the HIV dataset against WCE. **Table S9.** P-values of the Welch tests (N = 50) for the Tox21 dataset against WCE. **Table S10.** P-values of the Welch tests (N = 50) for the MUV dataset against WCE. **Table S11.** P-values of the Welch tests (N = 5) for the Phosphatase dataset against WCE. **Table S12.** P-values of the Welch tests (N = 5) for the NTPase dataset against WCE. **Table S13.** P-values of the Welch tests (N = 50) for the datasets from the MoleculeNet repository against the models from Arshadi et al. **Table S14.** Summary of the benchmarking results for the HTS dataset. The best values for each metric in each dataset are highlighted in bold. **Table S15.** P-values of the Welch tests (N = 5) for the HTS dataset against WCE. **Table S16.** Boosting iterations for each loss function with optimal hyperparameters for the HIV dataset. **Table S17.** Boosting iterations for each loss function with optimal hyperparameters for the HIV dataset.

Acknowledgements

The authors thank Isabel Wilkinson for her assistance in proofreading and revising the manuscript and Beatrix Blume and Phil Hewitt for setting up and performing the HTS assay and discussing the assay results.

Author contributions

DB ideated the study, wrote the code, executed the benchmarks on the MolData and MoleculeNet datasets and wrote the manuscript. LF executed the benchmark on the industrial dataset, helped to design the study and contributed to the manuscript preparation. DK helped to design the study and contributed to the manuscript preparation. SAS helped to design the study and contributed to the manuscript preparation. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. The authors thank Merck KGaA Darmstadt for their generous support with the Merck Future Insight Prize 2020.

Availability of data and materials

The full data and the code required to reproduce the results described in this study are available at the following github repository: https://github.com/dahvida/gradient_boosting_CLF.

Declarations

Competing interests

The authors declare no competing financial interest.

Author details

[1]Center for Functional Protein Assemblies, Technical University of Munich (TUM), Ernst-Otto-Fischer-Straße 8, 85784 Garching, Germany. [2]Merck Healthcare KGaA, Frankfurter Straße 250, 64293 Darmstadt, Germany.

Received: 2 September 2022 Accepted: 30 October 2022
Published online: 10 November 2022

References

1. Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, Li B, Madabhushi A, Shah P, Spitzer M, Zhao S (2019) Applications of machine learning in drug discovery and development. Nat Rev Drug Discov 18(6):463–477. https://doi.org/10.1038/s41573-019-0024-5
2. Sripriya Akondi V, Menon V, Baudry J, Whittle J (2022) Novel big data-driven machine learning models for drug discovery application. Molecules 27(3):594. https://doi.org/10.3390/molecules27030594
3. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE (2021) PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res 49(D1):D1388–D1395. https://doi.org/10.1093/nar/gkaa971
4. Irwin JJ, Tang KG, Young J, Dandarchuluun C, Wong BR, Khurelbaatar M, Moroz YS, Mayfield J, Sayle RA (2020) ZINC20—a free ultralarge-scale chemical database for ligand discovery. J Chem Inf Model 60(12):6065–6073. https://doi.org/10.1021/acs.jcim.0c00675
5. Korkmaz S (2020) Deep learning-based imbalanced data classification for drug discovery. J Chem Inf Model 60(9):4180–4190. https://doi.org/10.1021/acs.jcim.9b01162
6. Schneider P, Müller AT, Gabernet G, Button AL, Posselt G, Wessler S, Hiss JA, Schneider G (2017) Hybrid network model for "deep learning" of chemical data: application to antimicrobial peptides. Mol Inform 36(1–2):1600011. https://doi.org/10.1002/minf.201600011
7. Stokes JM, Yang K, Swanson K, Jin W, Cubillos-Ruiz A, Donghia NM, MacNair CR, French S, Carfrae LA, Bloom-Ackermann Z, Tran VM, Chiappino-Pepe A, Badran AH, Andrews IW, Chory EJ, Church GM, Brown ED, Jaakkola TS, Barzilay R, Collins JJ (2020) A deep learning approach to antibiotic discovery. Cell 180(4):688-702.e13. https://doi.org/10.1016/j.cell.2020.01.021
8. Gawriljuk VO, Foil DH, Puhl AC, Zorn KM, Lane TR, Riabova O, Makarov V, Godoy AS, Oliva G, Ekins S (2021) Development of machine learning models and the discovery of a new antiviral compound against yellow fever Virus. J Chem Inf Model. https://doi.org/10.1021/acs.jcim.1c00460
9. Chuang KV, Gunsalus LM, Keiser MJ (2020) Learning molecular representations for medicinal chemistry: miniperspective. J Med Chem 63(16):8705–8722. https://doi.org/10.1021/acs.jmedchem.0c00385

10. Winter R, Montanari F, Noé F, Clevert D-A (2019) Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. Chem Sci 10(6):1692–1701. https://doi.org/10.1039/C8SC04175J
11. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A (2018) Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent Sci 4(2):268–276. https://doi.org/10.1021/acscentsci.7b00572
12. Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, Guzman-Perez A, Hopper T, Kelley B, Mathea M, Palmer A, Settels V, Jaakkola T, Jensen K, Barzilay R (2019) Analyzing learned molecular representations for property prediction. J Chem Inf Model 59(8):3370–3388. https://doi.org/10.1021/acs.jcim.9b00237
13. Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, Shen C, Cao D, Wu J, Hou T (2021) Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. J Cheminformatics 13(1):12. https://doi.org/10.1186/s13321-020-00479-8
14. Altae-Tran H, Ramsundar B, Pappu AS, Pande V (2017) Low data drug discovery with one-shot learning. ACS Cent Sci 3(4):283–293. https://doi.org/10.1021/acscentsci.6b00367
15. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) MoleculeNet: a benchmark for molecular machine learning. Chem Sci 9(2):513–530. https://doi.org/10.1039/C7SC02664A
16. Siramshetty VB, Nguyen D-T, Martinez NJ, Southall NT, Simeonov A, Zakharov AV (2020) Critical analysis. J Chem Inf Model 60(12):6007–6019. https://doi.org/10.1021/acs.jcim.0c00884
17. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global Understanding with explainable AI for trees. Nat Mach Intell 2(1):56–67. https://doi.org/10.1038/s42256-019-0138-9
18. Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. https://doi.org/10.48550/arXiv.1705.07874
19. Shwartz-Ziv R, Armon A (2022) Tabular data: deep learning is not all you need. Inf Fusion 81:84–90. https://doi.org/10.1016/j.inffus.2021.11.011
20. Keshavarzi Arshadi A, Salem M, Firouzbakht A, Yuan JS (2022) MolData, a molecular benchmark for disease and target based machine learning. J Cheminformatics 14(1):10. https://doi.org/10.1186/s13321-022-00590-y
21. Esposito C, Landrum GA, Schneider N, Stiefl N, Riniker S (2021) GHOST: Adjusting the decision threshold to handle imbalanced data in machine learning. J Chem Inf Model 61(6):2623–2640. https://doi.org/10.1021/acs.jcim.1c00160
22. HaiboHeGarcia EA (2009) Learning from Imbalanced Data. IEEE Trans. Knowl. Data Eng. 21(9):1263–1284
23. Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. Int J Pattern Recognit Artif Intell 23(04):687–719. https://doi.org/10.1142/S0218001409007326
24. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2018) Focal loss for dense object detection. ArXiv170802002 42(2):318–327
25. Cao K, Wei C, Gaidon A, Arechiga N, Ma T (2019) Learning imbalanced datasets with label-distribution-aware margin loss. ArXiv. https://doi.org/10.48550/arXiv.1906.07413
26. Tan J, Wang C, Li B, Li Q, Ouyang W, Yin C, Yan J (2020) Equalization loss for long tailed object recognition. ArXiv:200305176 arXiv. https://doi.org/10.4550/arXiv.2003.05176
27. Menon AK, Jayasumana S, Rawat AS, Jain H, Veit A, Kumar S (2021) Long-tail learning via logit adjustment. arXiv 9:07314
28. Casanova-Alvarez O, Morales-Helguera A, Cabrera-Pérez MÁ, Molina-Ruiz R, Molina C (2021) A novel automated framework for QSAR modeling of highly imbalanced *Leishmania* high-throughput screening data. J Chem Inf Model 61(7):3213–3231. https://doi.org/10.1021/acs.jcim.0c01439
29. Idakwo G, Thangapandian S, Luttrell J, Li Y, Wang N, Zhou Z, Hong H, Yang B, Zhang C, Gong P (2020) Structure-activity relationship-based chemical classification of highly imbalanced Tox21 datasets. J Cheminform 12(1):66. https://doi.org/10.1186/s13321-020-00468-x
30. Yuchun Tang; Yan-Qing Zhang. Granular SVM with Repetitive undersampling for highly imbalanced protein homology prediction. In *2006 IEEE International Conference on Granular Computing*; IEEE: Atlanta, 2006 457–460. https://doi.org/10.1109/GRC.2006.1635839
31. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. in *proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining*; ACM: San Francisco California, 2016 785–794. https://doi.org/10.1145/2939672.2939785
32. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: 2017. A highly efficient gradient boosting decision tree in. *Advances in neural information processing systems*; curran associates, Inc 30
33. Jiménez-Luna J, Grisoni F, Schneider G (2020) Drug discovery with explainable artificial intelligence. Nat Mach Intell 2(10):573–584. https://doi.org/10.1038/s42256-020-00236-4
34. Dahlin JL, Nissink JWM, Strasser JM, Francis S, Higgins L, Zhou H, Zhang Z, Walters MA (2015) PAINS in the assay: chemical mechanisms of assay interference and promiscuous enzymatic inhibition observed during a sulfhydryl-scavenging HTS. J Med Chem 58(5):2091–2113. https://doi.org/10.1021/jm5019093
35. David L, Walsh J, Sturm N, Feierberg I, Nissink JWM, Chen H, Bajorath J, Engkvist O (2019) Identification of compounds that interfere with high-throughput screening assay technologies. ChemMedChem 14(20):1795–1802. https://doi.org/10.1002/cmdc.201900395
36. Friedman JH (2001) greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
37. Biau G, Scornet E (2016) A random forest guided tour. TEST 25(2):197–227. https://doi.org/10.1007/s11749-016-0481-7
38. Wang C, Deng C, Wang S (2020) Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. Pattern Recognit Lett 136:190–197. https://doi.org/10.1016/j.patrec.2020.05.035
39. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2017) CatBoost: unbiased boosting with categorical features. https://doi.org/10.48550/arXiv.1706.09516
40. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. https://doi.org/10.1007/BF00994018
41. Chicco D, Jurman G (2020) The Advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1):6. https://doi.org/10.1186/s12864-019-6413-7
42. Feng Y, Zhou M, Tong X (2021) Imbalanced classification: a paradigm-based review. arXiv 14:383–406
43. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD (2015) Hyperopt: a python library for model selection and hyperparameter optimization. Comput Sci Discov 8(1):014008. https://doi.org/10.1088/1749-4699/8/1/014008
44. Zhang Y-F, Ren W, Zhang Z, Jia Z, Wang L, Tan T (2022) Focal and efficient IOU loss for accurate bounding box regression. Neurocomputing 506:146–157. https://doi.org/10.1016/j.neucom.2022.07.042
45. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D (2020) Distance-IoU loss: faster and better learning for bounding box regression. Proc AAAI Conf Artif Intell 34(07):12993–13000. https://doi.org/10.1609/aaai.v34i07.6999
46. Yeung M, Sala E, Schönlieb C-B, Rundo L (2021) Unified focal loss: generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. arXiv 24:102026
47. Ustimenko A, Prokhorenkova L (2021) SGLB: Stochastic Gradient Langevin Boosting. https://doi.org/10.48550/arXiv.2001.07248

Publisher's Note

B.

Paper 2 (chapter 4)

Accepted open access article in *Journal of Cheminformatics* 15, 73 (**2023)**.

by **Davide Boldini**, Francesca Grisoni, Daniel Kuhn, Lukas Friedrich and Stephan A. Sieber.

https://doi.org/10.1186/s13321-023-00743-7

Journal of Cheminformatics

RESEARCH

Open Access

Practical guidelines for the use of gradient boosting for molecular property prediction

Davide Boldini[1], Francesca Grisoni[2,3], Daniel Kuhn[4], Lukas Friedrich[4] and Stephan A. Sieber[1*]

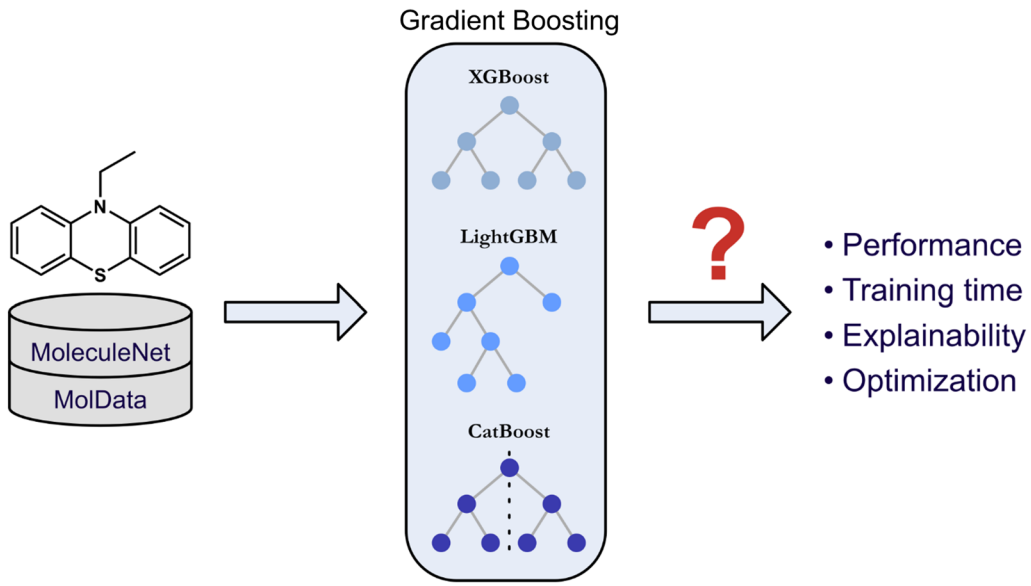
Abstract

Decision tree ensembles are among the most robust, high-performing and computationally efficient machine learning approaches for quantitative structure–activity relationship (QSAR) modeling. Among them, gradient boosting has recently garnered particular attention, for its performance in data science competitions, virtual screening campaigns, and bioactivity prediction. However, different variants of gradient boosting exist, the most popular being XGBoost, LightGBM and CatBoost. Our study provides the first comprehensive comparison of these approaches for QSAR. To this end, we trained 157,590 gradient boosting models, which were evaluated on 16 datasets and 94 endpoints, comprising 1.4 million compounds in total. Our results show that XGBoost generally achieves the best predictive performance, while LightGBM requires the least training time, especially for larger datasets. In terms of feature importance, the models surprisingly rank molecular features differently, reflecting differences in regularization techniques and decision tree structures. Thus, expert knowledge must always be employed when evaluating data-driven explanations of bioactivity. Furthermore, our results show that the relevance of each hyperparameter varies greatly across datasets and that it is crucial to optimize as many hyperparameters as possible to maximize the predictive performance. In conclusion, our study provides the first set of guidelines for cheminformatics practitioners to effectively train, optimize and evaluate gradient boosting models for virtual screening and QSAR applications.

Keywords Gradient boosting, Virtual screening, QSAR

*Correspondence:
Stephan A. Sieber
stephan.sieber@tum.de
Full list of author information is available at the end of the article

Graphical abstract



Introduction

Quantitative structure–activity relationship (QSAR) modelling occupies a vital role in cheminformatics research [1–5]. QSAR aims to link the molecular structure with experimentally measurable properties, and it is routinely used to predict molecular properties such as bioactivity [6–9], toxicity [10–13] and absorption, distribution, metabolism and excretion (ADME) [3, 14], thus covering a fundamental role in both hit discovery and hit-to-lead optimization.

QSAR aims to link the molecular structure (numerically encoded as the so-called molecular descriptors) [15–17] with experimentally measurable properties. For this application, decision tree ensembles are among the most used machine learning methods thanks to their excellent performance, ability to rank features in terms of importance and their ability to scale to large datasets [18, 19], alongside other popular frameworks like support vector machines (SVM) [20, 21].

Among decision tree ensembles, gradient boosting machines (GBM) have seen a strong surge in popularity in the last years, driven by excellent results in data science competitions and state-of-the-art performance in modelling tabular data [22, 23]. GBM iteratively aggregates predictive models so that each one compensates the errors from the previous step, thus yielding a high-performance ensemble.

In cheminformatics, GBM has already found widespread use in several QSAR tasks such as toxicity prediction [12], drug sensitivity analysis [24], anti-cancer activity modelling [25] and drug-target interaction identification [26], as well as showing competitive performance with deep learning approaches in recent large-scale benchmarking studies [16, 27–30].

However, several implementations of the GBM algorithm exist, each with unique modifications to the original formulation and employing different decision tree structures [23], such as XGBoost [31], LightGBM [32] and CatBoost. [33] While the importance of these differences has been recognised in other fields [23], these algorithms are used interchangeably in chemoinformatics, and to our knowledge their respective advantages are not well documented. Thus, there is an urgent need for a rigorous benchmarking of these different implementations for QSAR applications. This is further warranted by the uniqueness of cheminformatics datasets compared to other typical tabular datasets like finance and real estate price prediction [22, 23]. For example, datasets in this field tend to have a much higher number of features, they are often extremely imbalanced [34] and might contain false positives or false negatives [35].

The aim of this paper is to provide the first set of practical guidelines for the use of gradient boosting in QSAR applications, such as toxicology and drug discovery, by answering the following questions:

1. Which gradient boosting implementation performs the best for QSAR?

2. Which package scales the best to large datasets, such as high throughput screens (HTS)?
3. Do they produce similar feature importance rankings, or do they highlight different molecular features?
4. Is it possible to identify the most important hyperparameters to optimize for these algorithms to accelerate further the development and deployment of these methods for QSAR?

To answer these questions, we carried out a large-scale benchmark of these three implementations on 16 classification and regression datasets with 94 different endpoints commonly considered for virtual screening, covering a wide range of dataset size and class-imbalance ratios. To ensure the robustness of our results, we extensively optimized each algorithm according to the guidelines set up by the respective authors of the packages and recent studies, constructing 157,590 individual QSAR models.

Methods

GBM is an ensemble algorithm, which aims to aggregate several decision trees into a single more performant predictor. Decision trees are a machine-learning algorithm that learns a flowchart-like structure of hierarchical binary decisions [36]. The terminal nodes of the graph are generally named leaves, which are used to assign sample predictions [36]. To explain how GBM constructs the decision tree ensemble, we first present the original implementation of the algorithm [37] followed by a systematic analysis of the changes introduced by XGBoost, LightGBM and CatBoost.

Gradient boosting

Given an input matrix X and a vector Y of molecular properties (e.g., biological activity), the gradient boosting algorithm approximates the underlying function $F(x)$, which maps the relationship between the molecular descriptor x_i and the biological activity y_i, with a function $\hat{F}(x)$, constructed in an additive manner:

$$\hat{F}(x) = \sum_{m=1}^{M} \sigma * \widehat{F_m}(x) \tag{1}$$

where σ is the learning rate, a constant regularization parameter limiting the influence of a given predictor within the ensemble, and $\widehat{F_m}(x)$ is the mth tree. Given a loss function $L\left(y_i, p_i\right)$, such as the binary cross-entropy, that measures the quality of predictions p_i with respect to real readouts y_i, after the first iteration each new tree $\widehat{F_m}$ is learned by minimizing the following objective:

$$\widehat{F_m} = argminE\left(\frac{-\partial L\left(Y, P_{m-1}\right)}{\partial P_{m-1}} - P_m\right) \tag{2}$$

where the derivative of the loss with respect to the ensemble output represents the prediction residuals of $\hat{F}(x)$ at the previous iteration, and P_m are the predictions at the current iteration. As such, each new decision tree is constructed so that it compensates the prediction errors of the model during the previous iteration, essentially conducting gradient descent in function space instead of parameter space.

The original formulation of GBM is the one employed by the popular machine learning package Scikit-learn [38]. Unfortunately, this implementation lacks many of the regularization and optimization methods implemented by XGBoost, CatBoost and LightGBM and cannot be parallelized across multiple CPU cores. For this reason, we did not include the Scikit-learn version of GBM in the benchmarking study.

XGBoost

XGBoost introduces a regularized learning objective [31]. At a given iteration m, instead of being computed according to the loss function $L\left(y_i, p_i\right)$, the residuals are calculated with the following formula:

$$L_{\emptyset}\left(y, p\right) = \sum_{i=1}^{I} L\left(y_i, p_i\right) + \gamma T_m + \frac{1}{2}\lambda \|w_m\|^2 \tag{3}$$

where γ and λ are regularization hyperparameters, T_m is the number of leaves in the mth tree and $\|w_m\|^2$ is the L2 norm of its leaf weights. Thanks to this modification, XGBoost learns simpler trees with smoother weights, which leads to better generalization [31]. Additionally, XGBoost employs Newton descent instead of gradient descent to optimize its trees, which leads to faster convergence [39]. Finally, XGBoost also introduced a new feature split finding algorithm to speed up training [31].

LightGBM

This implementation also adopts many solutions proposed by XGBoost to improve the performance such as the regularized learning objective and Newton descent. However, LightGBM introduces three new strategies to make training more efficient: a histogram-based split finding method, Exclusive Feature Bundling (EFB) and Gradient-based One-Side Sampling (GOSS) [32]. EFB employs heuristics to find groups of mutually exclusive features and merges them together, thus reducing the dimensionality of the dataset, while GOSS relies on gradients to sample at each iteration the most important dataset instances without changing the training set distribution. Each of these algorithms simplifies different aspects of the original minimization objective, thus

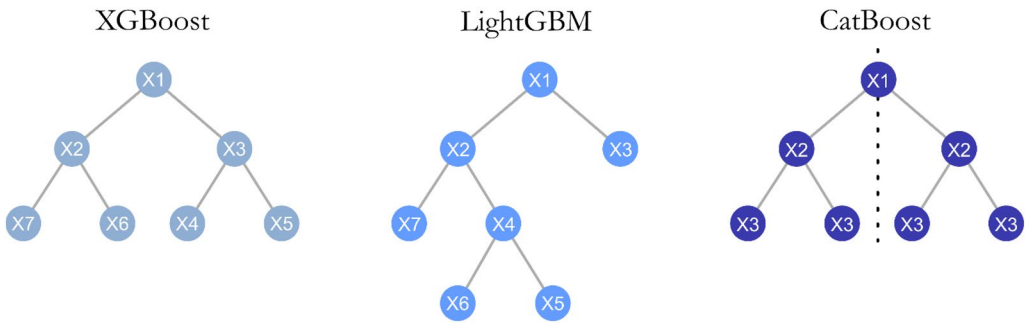


Fig. 1 Different tree structures and split indexes (shown inside each node) generated by XGBoost, LightGBM and CatBoost. XGBoost adopts a "breadth-first" search, maintaining constant tree depth across branches. LightGBM uses a "depth-first" criterion, yielding asymmetric trees. CatBoost relies on oblivious trees, where at a given depth the same split is used across all branches, as indicated by the constant split indexes

speeding up training time with negligible loss in accuracy. Furthermore, LightGBM employs a different tree growth strategy compared to XGBoost. In most cases, trees are generated in a "breadth-first" fashion, where every time a new split is found, all other splits at the same level are first considered before increasing further the depth of the tree. This yields tree structures that have the same depth across all branches. In contrast, LightGBM grows trees in a "depth-first" fashion (Fig. 1), where the algorithm splits nodes exclusively according to the largest performance gain [40]. This procedure leads to asymmetric trees, where certain branches might be very deep while others might be shallow. This approach tends to converge faster, but might be susceptible to overfitting on small datasets [32].

Catboost

There are three main features that distinguish CatBoost from LightGBM and XGBoost. First, it provides a novel Target Statistics (TS) algorithm to handle categorical variables, which leads to more robust performance on unseen data by addressing the issue of target leakage during training [33]. However, categorical inputs are very rarely found in molecular descriptors [41], therefore this aspect is not of big relevance for cheminformatics applications. Second, it introduced ordered boosting, a variation of gradient boosting where each model is trained on a different partition of the training dataset, tackling the issue of prediction shift that arises by fitting trees on gradients obtained from samples already used during training. In principle, this approach reduces the risk of overfitting, especially on small datasets [33]. Third, CatBoost employs "oblivious decision trees", where the same variable and threshold are used to generate splits at a given depth level (Fig. 1) [33, 42]. This enforced symmetry acts as regularization, constraining the expressiveness of tree models, and can be leveraged to provide uncertainty estimates on predictions, similarly to Gaussian Processes models [43]. Finally, the authors of this library have researched extensively the theoretical properties of gradient boosting and proposed several new features like Langevin gradient descent [44] and sample importance analysis [45], which are only available in the CatBoost package [42].

Experiments

Datasets

To provide a robust evaluation framework for our benchmark analysis, we evaluated XGBoost, LightGBM and CatBoost on 16 classification and regression datasets from three well-established repositories: MoleculeNet, [27] MolData [1] and the ChEMBL benchmarking study from Cortés-Ciriano et al [46] (Table 1). From the first, we included Tox21, MUV, HIV, ClinTox, BBBP, BACE and SIDER. From the second, we chose the Phosphatase, NTPase, Oxidoreductase and Fungal datasets. From the third, we selected HERG, Acetylcholinesterase, COX-2, erbB1 and JAK-2. We retrieved the MoleculeNet datasets from a recent benchmarking study [16], while we referred to the original publications for the MolData repository and the ChEMBL datasets [1, 46]. This selection entails approximately 1.4 million unique compounds and 94 endpoints on a wide variety of protein families and biological responses, ensuring that our findings are broadly applicable for cheminformatics applications. Our selection covers an extensive range of compounds per endpoint (from 2000 to 330,000) and imbalance ratios between compounds classified as either 'positive' or 'negative' (from 1:2 to 1:500), reflecting the diversity of datasets typically encountered in cheminformatics (Table 1).

Performance metrics

For each classification dataset, we evaluated the Receiver Operating Characteristic Area Under Curve (ROC-AUC) and Precision-Recall Area Under Curve (PR-AUC). Our selection is consistent with the figures of merit used in

Table 1 Datasets employed in this study. For datasets with multiple endpoints, we reported the ranges between minimum and maximum values regarding the compounds per endpoint and imbalance ratios

Name	Type	Source	Endpoints	Compounds per endpoint	Class imbalance ratio
Tox21	Classification	MoleculeNet	12	5810–7265	1:5–1:33
HIV	Classification	MoleculeNet	1	40,748	1:27
MUV	Classification	MoleculeNet	17	14,415–14,903	1:486–1:613
BACE	Classification	MoleculeNet	1	1513	1:1
BBBP	Classification	MoleculeNet	1	2039	1:3
SIDER	Classification	MoleculeNet	27	1427	1:12–1:63
ClinTox	Classification	MoleculeNet	2	1478	1:12–1:14
Phosphatase	Classification	MolData	5	260,322–298,215	1:121–1:576
NTPase	Classification	MolData	6	251,895–301,932	1:3–1:16,265
Oxidoreductase	Classification	MolData	10	79,853–325,083	1:9–1:9847
Fungal	Classification	MolData	7	152,880–302,256	1:135–1:640
HERG	Regression	Cortés-Ciriano et al.	1	5207	N.A
Acetylcholinesterase	Regression	Cortés-Ciriano et al.	1	3159	N.A
COX-2	Regression	Cortés-Ciriano et al.	1	2855	N.A
erbB1	Regression	Cortés-Ciriano et al.	1	4868	N.A
JAK-2	Regression	Cortés-Ciriano et al.		2655	N.A

the literature when evaluating these datasets and ensures that the results are not skewed by high imbalance ratios [1, 16, 27, 47, 48]. For the regression datasets, we evaluated the Root Mean Squared Error (RMSE). To assess whether differences in performance are statistically significant, we used the two-tailed Mann–Whitney test with Bonferroni correction [49].

Molecular descriptors

We featurized all compounds using the Extended-Connectivity Fingerprints (ECFP) with radius of 2 and bit size of 1024 [50]. To ensure that bit collision is not a factor in any of our findings, we have investigated the change in vector sparsity when using larger bit sizes. Given that the number of unique fragments remains approximately constant for all datasets when increasing the bit size (Additional file 1: Table S1), we can exclude that bit collision plays a role for the benchmarks in this study.

Performance analysis

We used three different optimization and evaluation protocols, depending on whether the dataset is from MoleculeNet, MolData or ChEMBL. The reason for this is to keep our analysis consistent with prior studies from the scientific literature, and because the datasets from MolData are several orders of magnitude larger than the ones in the MoleculeNet repository or from Cortés-Ciriano et al [46].

For MoleculeNet datasets, we replicated a previously proposed procedure [16], whereby for each endpoint, each classifier is optimized with Hyperopt [51] for 100 iterations using an extensive hyperparameter grid, determined according to existing guidelines and benchmarks [22, 39, 40, 42]. The full hyperparameter grid is available in the Supporting Information. Each optimization iteration measured the average PR-AUC with a given hyperparameter setting across three random train-test splits with an 80:20 ratio. Then, the model was run with the optimal hyperparameters on 50 independent evaluations with random splits, using the same ratio between training and test set. After each run, the ROC-AUC and PR-AUC were measured on the test set as well as the training time. Finally, for a given dataset, the performance metrics and training times were averaged across replicates and across endpoints.

For the MolData benchmarks, we used the scaffold splits provided by Arshadi and coworkers during optimization and evaluation [1]. As such, for each endpoint, each classifier was optimized for 100 iterations using Hyperopt [51] with the same grid as above. Each iteration measured the PR-AUC obtained by the classifier with a given hyperparameter setting on the validation set. Then, the model was run with optimal hyperparameters on five independent evaluations with different random seeds, measuring the ROC-AUC and PR-AUC on the test set as well as the training time. As above, the results were reported as averages across replicates and endpoint for a given dataset.

For the regression datasets from Cortés-Ciriano et al. [46], we adopted the procedure employed in the original publication. In short, each dataset was split into training, validation and test sets with a 70:15:15 ratio using

random splits. We then performed hyperparameter tuning via Hyperopt, optimizing RMSE on the validation split for 100 iterations, using the same grid as above. Finally, we repeated training on the training split and evaluation of RMSE on the test set for 50 iterations. As such, the final RMSE values were indicated as averages across replicates for each dataset.

Feature ranking analysis

One of the advantages of GBM is that it can provide information on the feature importance, which can be used as a tool to provide indication of what drives the model predictions, and, in certain cases, to achieve model explainability [52]. We used Shapley values [19, 53] to assess which molecular features are the most important according to each GBM predictor. Shapley values quantify the importance of each feature ('feature attribution' [37]) by evaluating the change in a model's predictions across all possible permutations [19, 52]. To obtain feature rankings for each dataset, we collected the Shapley values from each model with optimal hyperparameters during the evaluation procedure. Then, we averaged them across independent runs and dataset endpoint, obtaining one ranked list of variables per dataset for each model. To compare the variable rankings between pairs of GBM implementations, we employed the following formula:

$$Overlap\% = \left(1 - \frac{V_{sk}}{k}\right) * 100, k = 20 \quad (4)$$

where k is the cut-off for the number of most important variables to consider (set to $k=20$ in the present study) and V_{sk} is the number of unique variables when considering both importance rankings. Intuitively, this metric measures the agreement of the two rankings, irrespective of the specific ordering, among the top 20 most important variables. For example, a score of 50 indicates that two GBM models have 10 molecular features in common when looking at their respective top 20 most important variables, regardless of whether these 10 features received the same rank in both lists. This score therefore shows whether the use of different gradient boosting algorithms would highlight the same features as most important, without being influenced by the ranking of less informative variables. However, it should be kept in mind that for many molecular representations such as hashed fingerprints, translating feature importance rankings into chemical insights is not a trivial task [54].

Finally, to evaluate the influence of converging to different hyperparameter configurations, regardless of algorithmic differences in the gradient boosting implementation, we also evaluated the feature ranking overlap between two independent LightGBM optimization runs. The analysis was limited to LightGBM due to computational costs and that considering one GBM is sufficient to evaluate the variability in feature ranking overlap induced by the stochasticity in the hyperparameter optimization process.

Hyperparameter analysis

To evaluate the influence of each hyperparameter on the optimization process, we employed the Functional ANOVA (fANOVA) [55]. To acquire a sufficient collection of hyperparameter combinations, we optimized LightGBM with Hyperopt for 500 iterations on each endpoint, using the same hyperparameter grid and evaluation criteria as above. Because of the high computational cost for this analysis, we limited our study only to one GBM implementation and exclusively to classification datasets. Then, after pruning the worst 150 iterations, we processed the resulting parameter-performance pairs using fANOVA, yielding individual hyperparameter importance scores and their first-order interactions. By limiting the analysis to well-performing configurations, we ensured that the importance estimates for the parameters reflect their importance on reaching the optimum, and not on causing large oscillations in performance [55]. We excluded the SIDER and Fungal datasets from this analysis, since they were reserved as test sets to evaluate whether selecting hyperparameters according to their fANOVA importance score generalizes to unseen datasets. Furthermore, to assess the influence of molecular descriptors on the optimal hyperparameters, we also repeated this procedure using the MACCS keys [56] and an assortment of 207 physical–chemical descriptors from RDKit as featurization options. The complete list of descriptors is available in the Supporting Information (Additional file 1: Table S2).

Software and implementation

Molecular descriptors were computed using RDKit (Version 2022.09.4) for python. [50] For training the models, XGBoost (Version 1.7.1) [39], LightGBM (Version 3.3.5) [40] and CatBoost (Version 1.1.1) [42] were employed. Scikit-learn (Version 1.2.1) [38] was used to split the MoleculeNet datasets and compute ROC-AUC and PR-AUC values. Each model was tuned via Bayesian hyperparameter optimization using the Hyperopt package (Version 0.2.7) [51]. Finally, SHAP (Version 0.41.0) [19] was utilized to compute Shapley values and the fANOVA package (Version 2.0.5) [55] was employed for the hyperparameter importance analysis. All calculations were performed on an AMD Ryzen Threadripper 3970X CPU with 32 cores and 64 threads. Training of the gradient boosting models was parallelized across all cores available. The code to reproduce the results is available at https://github.com/dahvida/GBM_Benchmarking.

Results and discussion

Predictive performance

Overall, XGBoost achieves the best performance on most of the datasets (Fig. 2a, b and Additional file 1: Figure S1), with statistically significant differences in most cases. Interestingly, there seems to be a correlation between the improvement provided by XGBoost over the alternatives and dataset size. For smaller classification datasets (e.g., BACE, BBBP and ClinTox), CatBoost performs worse, with LightGBM being able to match or outperform XGBoost. This aspect is seemingly in contradiction with the concerns of overfitting due to its depth-first tree structure reported elsewhere. [40] For medium-sized datasets (e.g., Tox21, MUV and HIV, ranging from approximately 7000 compounds to 40,000), CatBoost tends to perform better than LightGBM, and it outperforms XGBoost on the Tox21 dataset. Finally, for large datasets (NTPase, Phosphatase and Oxidoreductase datasets, having more than 300,000 molecules per endpoint), XGBoost outperforms both LightGBM and CatBoost. When considering all datasets, XGBoost provides roughly a 5% improvement on average over LightGBM and CatBoost in terms of ROC-AUC and PR-AUC.

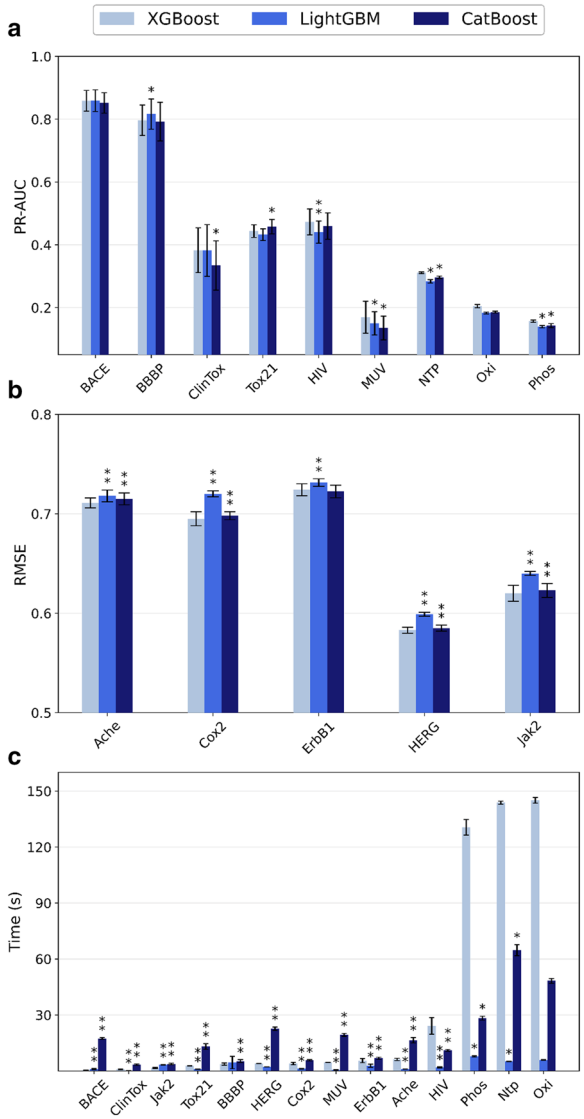


Fig. 2 Performance comparison of all gradient boosting implementations in terms of **a** PR-AUC, **b** RMSE and **c** training time. All calculations were performed on an AMD Ryzen Threadripper 3970X CPU. Statistical tests are carried out with respect to XGBoost. Error bars represent the standard deviation ($N = 50$ for MoleculeNet datasets, $N = 5$ for MolData datasets), while the asterisks denote whether the difference is significant (*: $\alpha < 0.05$, **: $\alpha < 0.01$, with Bonferroni correction)

Regarding the regression datasets, LightGBM tends to achieve worse RMSE scores, while XGBoost ranks as the best performing algorithm on most benchmarks (Fig. 2b). CatBoost is generally able to match the performance of XGBoost, although the differences are statistically significant.

When considering the training times across all datasets (Fig. 2c), a similar dependence on the dataset size can be observed. LightGBM is the fastest algorithm on all benchmarks, due to the algorithm's focus on reducing computational load. CatBoost is the slowest algorithm for small and medium sized datasets, while XGBoost requires significantly more time to train for larger datasets than both alternatives. While the absolute difference of training times for a single model is not particularly great (i.e., 5 versus 140 s on a CPU with 32 cores), it can significantly impact hyperparameter optimization procedures, where the model needs to be retrained many times. Furthermore, this difference will also grow significantly if less cores are available for training.

In summary, XGBoost provides the best predictive performance for cheminformatics out of all gradient boosting implementations, at the cost of training speed for larger datasets. LightGBM and CatBoost have comparable performance, but the former provides substantial benefits in terms of training time over the other algorithms.

Feature ranking comparison

We observed a remarkable variability between the importance rankings across different implementations, especially when comparing them to the overlap scores of two independent optimization and training runs for the same GBM algorithm (Fig. 3). For MUV, for example, there is approximately only a 20% overlap for any implementation pair, while for other datasets the agreement reaches up to 90%. The reason for the variability across

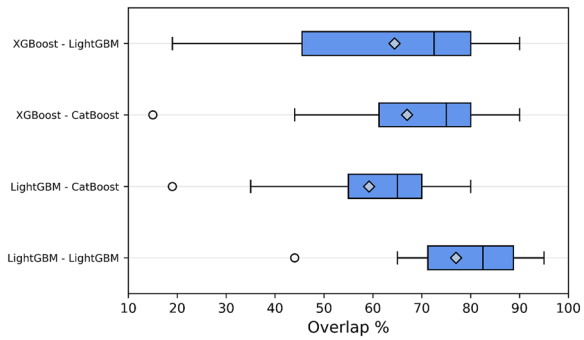


Fig. 3 Box-plot distribution of overlap scores across all datasets for each gradient boosting implementation pair. The length of the box denotes the interquartile range, the diamond indicates the mean and the horizontal line defines the median. The comparison between two independent optimization runs using the same algorithm was limited to LightGBM due to its computational cost

implementations could be due to the use of different tree structures, as well as converging to different hyperparameter optima. For example, tuning the minimum split gain can lead to the selection of different splits, which in turn would yield different variable importance scores. This would explain the results obtained when comparing two runs of the same GBM algorithm across all datasets, since even in that scenario the variable overlap scores are distributed between 70 and 90% (Fig. 3). Another possible explanation for this pattern is that the algorithms highlight similar molecular fragments, but those fragments are mapped to different bits in the ECFP representation, thus producing semantically similar rankings despite not focusing on the same variables. To investigate this hypothesis, we calculated the top 20 ranked fragments for all GBM algorithms for the BACE datasets and manually inspected them (Additional file 1: Figure S2). When comparing the most important fragments between pairs of GBM predictor, each model had approximately ten unique substructures, which did not have any analogues in the other rankings. As such, it seems that each implementation indeed generates semantically distinct explanations for a given dataset, highlighting potential differences in the learned structure–activity relationships.

The main takeaway from this analysis is that using gradient boosting to evaluate which molecular features or fragments are the most influential is a non-trivial task, given the low agreement between different implementations of the same algorithm. Expert knowledge must always be employed to evaluate each fingerprint bit or molecular descriptor and to assess whether the explanations provided by the model are reasonable. Finally, averaging the Shapley scores on different hyperparameter optima or across different gradient boosting implementations might yield better estimates of feature importance.

Hyperparameter importance

After calculating the hyperparameter importance across datasets for LightGBM, we evaluated their distribution on different endpoints (Fig. 4). The analysis was limited to one GBM implementation due to the high number of optimization iterations required per endpoint. We focused our analysis on the following hyperparameters:

- "*colsample_bytree*": fraction of features to sample at the beginning of the construction of a given tree. Tuning it helps with regularization of the ensemble.
- "*learning_rate*": regulates how much each tree affects the overall performance of the ensemble, or in other words how many boosting rounds are required to converge. Large learning rates help with underfitting, small learning rates can help with regularization.
- "*max_depth*": defines the maximum depth for constructing individual trees. Large values help with underfitting, small values can help with regularization.
- "*min_child_samples*": minimum number of samples for a given leaf node. Affects tree construction and can help with regularization.
- "*min_child_weight*": minimal sum of hessians for a given leaf node. Affects tree construction and can help with regularization.
- "*min_split_gain*": minimal decrease in loss required to further split a node. Affects tree construction and can help with regularization.
- "*neg_subsample*": fraction of majority class samples to use for bagging when constructing a given tree. Helps with class imbalance and regularization.
- "*num_leaves*": Maximum number of leaves a given tree can have. Similar to max_depth but provides more fine-grained control on the shape of the tree since LightGBM uses depth-first trees.
- "*reg_alpha*": L1 norm regularization coefficient of the leaf weights.
- "*reg_lambda*": L2 norm regularization coefficient of the leaf weights.
- "*scale_pos_weight*": scaling coefficient for the minority class when computing the cross-entropy loss. Large values can offset class imbalance.
- "*subsample*_freq": affects how often to perform bagging when training the ensemble. If set to *k*, bagging is performed every *k* trees.

Generally speaking, the importance of the individual hyperparameters in the optimization process varies greatly across datasets. Furthermore, 1st order

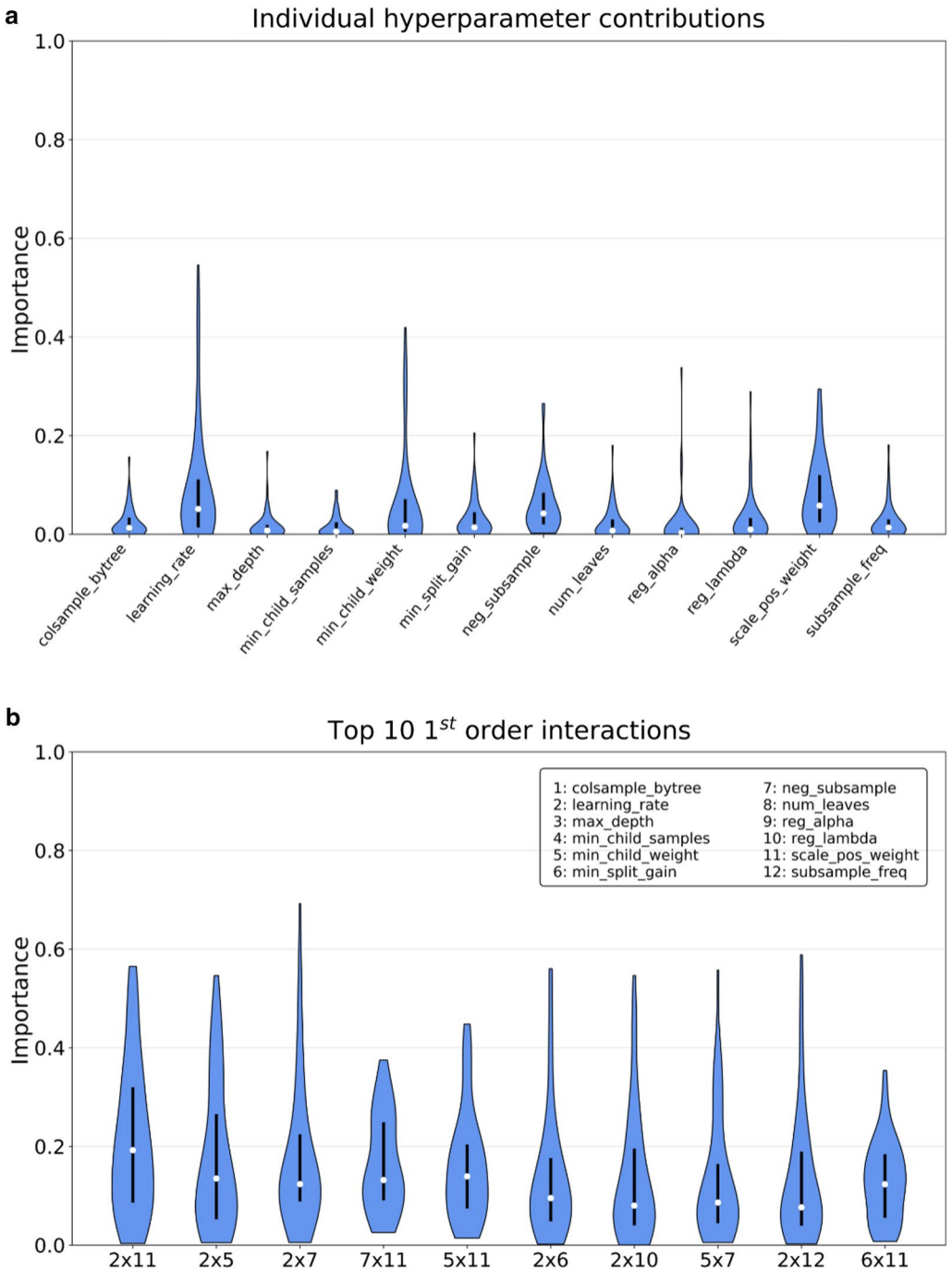


Fig. 4 Violin plot distribution of the importance scores across all endpoints for the Tox21, MUV, HIV, BBBP, BACE, ClinTox, Phosphatase, NTPase and Oxidoreductase datasets. **a**The distribution of individual contributions for each hyperparameter, denoted by a numerical identifier. **b** The score variation of pairwise interactions. Each interaction is defined by the combination of two numeric identifiers for conciseness

interactions between parameters play a more significant role in reaching the global optimum than tuning them in isolation, as highlighted by their larger importance score. This is consistent with the strong correlations between parameters and their non-linear effects on model behavior [39, 40, 42], which make Bayesian hyperparameter optimization necessary in the first place [51].

Looking at individual contributions (Fig. 4a), it is possible to identify highly influential hyperparameters, such as the learning rate and the minimum split gain, as well

as less relevant ones, such as tree-wise feature sampling. However, all importance score distributions are remarkably skewed, highlighting that each contribution can strongly vary across different datasets. When looking at the top ten most influential pairwise interactions (Fig. 4b), most of them are related to the learning rate and the scaling coefficient for the contribution of the minority class to the global loss, highlighting the importance of tuning weighted cross-entropy when dealing with imbalanced classification. While some of these findings are consistent with the optimization guidelines from the literature, such as tuning the learning rate and the minimum split gain, others appear to contradict them. For example, while stochastic sampling of instances and features is believed to be an effective regularization technique for gradient boosting [31], in this analysis tuning it seems to be not influential in converging to the parameter configuration optimum.

To evaluate the robustness of our importance estimates, we chose to optimize LightGBM again on all datasets, tuning only the most influential parameters according to the fANOVA analysis. To do so, we selected only the parameters that appeared at least once among the top 10 most important interaction terms, yielding a grid of 7 hyperparameters instead of 12 (available in the Supporting Information). To test whether this reduced selection leads to faster convergence of the optimization process, we used 30 iterations instead of 100. As a negative control, we also evaluated the performance achieved by optimizing all hyperparameters for the same number of iterations. Finally, we expressed the ROC-AUC and PR-AUC values achieved by these benchmarks as a fraction of the performance of the optimization process with all parameters and 100 iterations. This evaluation scheme allows us to assess how well quickly tuning only the most important hyperparameters approximates the original large-scale optimization procedure.

As shown in Fig. 5, given the same number of iterations, using only the best parameters for the optimization process leads to consistent performance gains compared to tuning all hyperparameters. This indicates that the scores from fANOVA accurately reflect the importance of tuning a given hyperparameter for reaching the optimum. Interestingly, in some cases the optimal hyperparameter grid is able to outperform the results obtained tuning all hyperparameters for 100 iterations, such as for the NTP dataset in terms of PR-AUC and ROC-AUC (Fig. 5 and Additional file 1: Figure S2).

However, when evaluating the effectiveness of adjusting only the most important parameters on holdout datasets, the performance improvements are inconsistent. This indicates that the hyperparameter importance scores obtained by analysis of a set of endpoints do not generalize on external endpoints (Additional file 1: Figure S1). Therefore, deciding which parameters to tune must be determined on a case-by-case basis. A similar pattern is also observed when evaluating the influence of changing molecular representation for constructing the QSAR

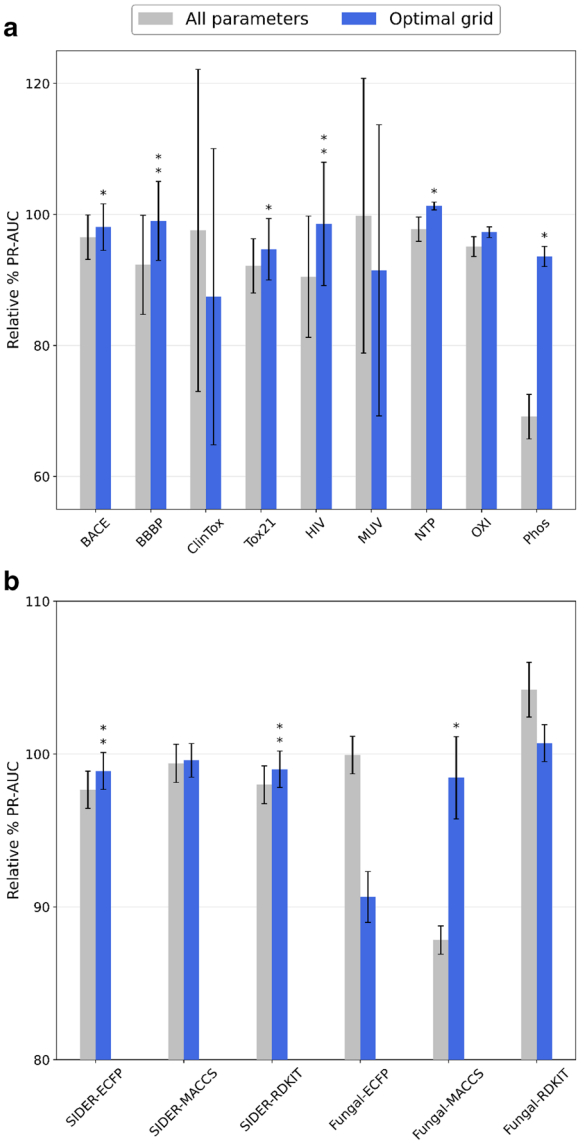


Fig. 5 LightGBM PR-AUC comparison between carrying out hyperparameter tuning according to the optimal grid obtained from fANOVA and tuning all hyperparameters. **a** Performance on the datasets used for the fANOVA analysis. **b** Performance on the holdout datasets and with different molecular representations. Each approach was optimized for 30 iterations. The performance is reported in relation to the results obtained by tuning all parameters for 100 iterations. Error bars represent the standard deviation ($N = 50$ for MoleculeNet datasets, $N = 5$ for MolData datasets), while the asterisks denote whether the difference is significant (*: $\alpha < 0.05$, **: $\alpha < 0.01$, with Bonferroni correction)

model, indicating that the parameter importance scores are highly feature-specific (Fig. 5 and Additional file 1: Figure S2).

In conclusion, optimization analysis tools such as fANOVA can be useful to further improve gradient boosting in cases where QSAR models need to be retrained periodically as new data is collected, for example for ADME prediction toolkits [3, 57]. However, the importance estimates provided by fANOVA do not generalize to unseen endpoints or different molecular representations, and limiting the optimization process to a handful of parameters can affect the performance of the classifier by up to 20%. Therefore, our recommendation is to tune all possible parameters when training gradient boosting models for QSAR, if the computational time to do so is not prohibitive. If optimizing all parameters is too costly, adjusting the learning rate, the weight of the minority class and the minimum gain to split will likely lead to the best results on a limited computational budget.

Conclusions

This work investigated the differences between popular gradient boosting implementations in the context of cheminformatics, to guide future QSAR modelling projects. Specifically, our analysis focused on predictive performance and training time, as well as on feature ranking consistency among methods. Furthermore, we investigated which hyperparameters are the most important to tune for gradient boosting machines to reach better performance faster. To achieve these goals, we evaluated 11 different datasets, encompassing approximately 1.4 million unique compounds with a diverse selection of dataset sizes and imbalance ratios.

XGBoost generally outperformed all alternatives in terms of predictive performance by approximately 5%, at the cost of longer training times for larger datasets (*e.g.* above 100,000 compounds). LightGBM and CatBoost achieve similar performance, but the former requires significantly less time to be trained compared to the other implementations. The improvement is especially significant for datasets with more than 100,000 compounds, where LightGBM could be trained approximately 100 times faster than XGBoost and 50 times faster than CatBoost. In terms of feature importance, each implementation tends to rank molecular features differently. This not only indicates that each approach might learn slightly different structure–activity relationships, but also that caution must be exercised when using these tools to assess which fragments or properties are relevant for the biological response modelled. In this context, expert knowledge is key to critically evaluate whether these explanations could be due to chance correlation. Finally, our hyperparameter importance analysis highlights that there is significant variability in how much a given parameter affects convergence to the optimum between datasets. As such, our indication is to tune as many parameters as possible when optimizing gradient boosting models. If the computational budget is limited, our recommendation is to focus on the learning rate, the minimum split gain and the weight of the minority class if the dataset is imbalanced.

In conclusion, our study provides a set of practical guidelines for the use of gradient boosting for molecular property prediction. Given the rising popularity of this algorithm for virtual screening and QSAR, we believe our study will provide useful advice in its optimization, its use cases and limitations, thus benefitting the cheminformatics community as a whole.

Abbreviations

QSAR	Quantitative structure-activity relationship
ADME	Absorption distribution metabolism excretion
SVM	Support vector machine
GBM	Gradient boosting machine
HTS	High throughput screening
EFB	Exclusive feature bundling
GOSS	Gradient-based one sided sampling
TS	Target Statistics
ROC-AUC	Receiver operator characteristic area under curve
PR-AUC	Precision recall area under curve
RMSE	Root mean squared error
ECFP	Extended connectivity fingerprint
fANOVA	Functional analysis of variance

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-023-00743-7.

Additional file 1: Figure S1. Performance comparison between classification models according to ROC-AUC. **Figure S2.** LightGBM ROC-AUC comparison between carrying out hyperparameter tuning according to the optimal grid obtained from fANOVA and tuning all hyperparameters. **a)** Performance on the datasets used for the fANOVA analysis. **b)** Performance on the holdout datasets and with different molecular representations. Each approach was optimized for 30 iterations. The performance is reported in relation to the results obtained by tuning all parameters for 100 iterations. Error bars represent the standard deviation (*N=50* for MoleculeNet datasets, *N=5* for MolData datasets), while the asterisks denote whether the difference is significant (*: $\alpha<0.05$, **: $\alpha<0.01$, with Bonferroni correction). **Figure S3.** Top 20 most important molecular fragments according to each GBM implementation for the BACE dataset. **Table S1.** Mean number of unique substructures per compound across datasets and bit sizes. **Table S2.** List of calculated 2D molecular descriptors from the RDKIT package.

Acknowledgements

The authors thank Günter Klambauer for the useful feedback on the project and the fruitful discussion on the results.

Author contributions

Conceptualization: D.B. Benchmarking: D.B. Software: D.B. Methodology: D.B. and F.G. Analysis of the results: all authors. Writing original draft: D.B. Writing review and editing: all authors. All authors have given approval to the final version of the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. The authors thank Merck KGaA Darmstadt for their generous support with the Merck Future Insight Prize 2020.

Availability of data and materials
The datasets and code supporting the conclusions of this article are available in the "GBM_Benchmarking" GitHub repository [https://github.com/dahvida/GBM_Benchmarking].

Declarations

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Bioscience, Center for Functional Protein Assemblies (CPA), Technical University of Munich, Garching bei Munich, Germany. [2]Department of Biomedical Engineering, Institute for Complex Molecular Sciences, Eindhoven University of Technology, Eindhoven, The Netherlands. [3]Centre for Living Technologies, Alliance TU/E, WUR, UU, UMC Utrecht, Utrecht, The Netherlands. [4]Merck Healthcare KGaA, Darmstadt, Germany.

Received: 31 March 2023 Accepted: 9 August 2023
Published online: 28 August 2023

References

1. Keshavarzi Arshadi A, Salem M, Firouzbakht A, Yuan JS (2022) MolData, a molecular benchmark for disease and target based machine learning. J Cheminf 14(1):10. https://doi.org/10.1186/s13321-022-00590-y
2. Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, Guzman-Perez A, Hopper T, Kelley B, Mathea M, Palmer A, Settels V, Jaakkola T, Jensen K, Barzilay R (2019) Analyzing learned molecular representations for property prediction. J Chem Inf Model 59(8):3370–3388. https://doi.org/10.1021/acs.jcim.9b00237
3. Aleksić S, Seeliger D, Brown JB (2021) ADMET Predictability at Boehringer Ingelheim: state-of-the-art, and do bigger datasets or algorithms make a difference? Mol Inform. https://doi.org/10.1002/minf.202100113
4. Mayr A, Klambauer G, Unterthiner T, Steijaert M, Wegner JK, Ceulemans H, Clevert D-A, Hochreiter S (2018) Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. Chem Sci 9(24):5441–5451. https://doi.org/10.1039/C8SC00148K
5. Chen H, Kogej T, Engkvist O (2018) Cheminformatics in drug discovery, an industrial perspective. Mol Inform 37(9–10):1800041. https://doi.org/10.1002/minf.201800041
6. Withnall M, Lindelöf E, Engkvist O, Chen H (2020) Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. J Cheminf 12(1):1. https://doi.org/10.1186/s13321-019-0407-y
7. Santana MVS, De S-J (2021) Novo design and bioactivity prediction of sars-cov-2 main protease inhibitors using recurrent neural network-based transfer learning. BMC Chem 15(1):8. https://doi.org/10.1186/s13065-021-00737-2
8. Gawriljuk VO, Zin PPK, Puhl AC, Zorn KM, Foil DH, Lane TR, Hurst B, Tavella TA, Costa FTM, Lakshmanane P, Bernatchez J, Godoy AS, Oliva G, Siqueira-Neto JL, Madrid PB, Ekins S (2021) Machine learning models identify inhibitors of SARS-CoV-2. J Chem Inf Model. https://doi.org/10.1021/acs.jcim.1c00683
9. Stokes JM, Yang K, Swanson K, Jin W, Cubillos-Ruiz A, Donghia NM, MacNair CR, French S, Carfrae LA, Bloom-Ackermann Z, Tran VM, Chiappino-Pepe A, Badran AH, Andrews IW, Chory EJ, Church GM, Brown ED, Jaakkola TS, Barzilay R, Collins JJ (2020) A deep learning approach to antibiotic discovery. Cell 180(4):688-702.e13. https://doi.org/10.1016/j.cell.2020.01.021
10. Jain S, Siramshetty VB, Alves VM, Muratov EN, Kleinstreuer N, Tropsha A, Nicklaus MC, Simeonov A, Zakharov AV (2021) Large-scale modeling of multispecies acute toxicity end points using consensus of multitask deep learning methods. J Chem Inf Model 61(2):653–663. https://doi.org/10.1021/acs.jcim.0c01164
11. Walter M, Allen LN, de la Vega de León A, Webb SJ, Gillet VJ (2022) Analysis of the benefits of imputation models over traditional QSAR models for toxicity prediction. J Cheminf 14(1):32. https://doi.org/10.1186/s13321-022-00611-w
12. Zhang J, Mucs D, Norinder U, Svensson F (2019) LightGBM: an effective and scalable algorithm for prediction of chemical toxicity-application to the tox21 and mutagenicity data sets. J Chem Inf Model. https://doi.org/10.1021/acs.jcim.9b00633
13. Grisoni F, Consonni V, Ballabio D (2019) Machine learning consensus to predict the binding to the androgen receptor within the CoMPARA project. J Chem Inf Model 59(5):1839–1848. https://doi.org/10.1021/acs.jcim.8b00794
14. Xiong G, Wu Z, Yi J, Fu L, Yang Z, Hsieh C, Yin M, Zeng X, Wu C, Lu A, Chen X, Hou T, Cao D (2021) ADMETlab 20: an integrated online platform for accurate and comprehensive predictions of ADMET properties. Nucleic Acids Res. https://doi.org/10.1093/nar/gkab255
15. Chuang KV, Gunsalus LM, Keiser MJ (2020) Learning molecular representations for medicinal chemistry: miniperspective. J Med Chem 63(16):8705–8722. https://doi.org/10.1021/acs.jmedchem.0c00385
16. Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, Shen C, Cao D, Wu J, Hou T (2021) Could Graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. J Cheminf 13(1):12. https://doi.org/10.1186/s13321-020-00479-8
17. Winter R, Montanari F, Noé F, Clevert D-A (2019) Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. Chem Sci 10(6):1692–1701. https://doi.org/10.1039/C8SC04175J
18. Biau G, Scornet E (2016) A random forest guided tour. TEST 25(2):197–227. https://doi.org/10.1007/s11749-016-0481-7
19. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2(1):56–67. https://doi.org/10.1038/s42256-019-0138-9
20. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. https://doi.org/10.1007/BF00994018
21. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A (2020) A comprehensive survey on support vector machine classification: applications. Chall Trends Neurocomp 408:189–215. https://doi.org/10.1016/j.neucom.2019.10.118
22. Shwartz-Ziv R, Armon A (2022) Tabular data: deep learning is not all you need. Inf Fusion 81:84–90. https://doi.org/10.1016/j.inffus.2021.11.011
23. Bentéjac C, Csörgő A, Martínez-Muñoz G (2021) A comparative analysis of gradient boosting algorithms. Artif Intell Rev 54(3):1937–1967. https://doi.org/10.1007/s10462-020-09896-5
24. Zheng S, Aldahdooh J, Shadbahr T, Wang Y, Aldahdooh D, Bao J, Wang W, Tang J (2021) Drugcomb update: a more comprehensive drug sensitivity data repository and analysis portal. Nucleic Acids Res 49(W1):W174–W184. https://doi.org/10.1093/nar/gkab438
25. Zhu Y, Brettin T, Evrard YA, Partin A, Xia F, Shukla M, Yoo H, Doroshow JH, Stevens RL (2020) Ensemble transfer learning for the prediction of anti-cancer drug response. Sci Rep 10(1):18040. https://doi.org/10.1038/s41598-020-74921-0
26. Zhang Y, Jiang Z, Chen C, Wei Q, Gu H, Yu B (2022) Deepstack-DTIs: predicting drug-target interactions using LightGBM feature selection and deep-stacked ensemble classifier. Interdiscip Sci Comput Life Sci 14(2):311–330. https://doi.org/10.1007/s12539-021-00488-7
27. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) MoleculeNet: a benchmark for molecular machine learning. Chem Sci 9(2):513–530. https://doi.org/10.1039/C7SC02664A
28. Siramshetty VB, Nguyen D-T, Martinez NJ, Southall NT, Simeonov A, Zakharov AV (2020) Critical analysis. J Chem Inf Model 60(12):6007–6019. https://doi.org/10.1021/acs.jcim.0c00884
29. Boldini D, Friedrich L, Kuhn D, Sieber SA (2022) Tuning gradient boosting for imbalanced bioassay modelling with custom loss functions. J Cheminf 14(1):80. https://doi.org/10.1186/s13321-022-00657-w

30. van Tilborg D, Alenicheva A, Grisoni F (2022) Exposing the limitations of molecular machine learning with activity cliffs. J Chem Inf Model 62(23):5938–5951. https://doi.org/10.1021/acs.jcim.2c01073
31. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM San Francisco California USA. 2016. https://doi.org/10.1145/2939672.2939785
32. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) LightGBM: a highly efficient gradient boosting decision tree in advances in neural information processing systems. Curran Assoc. https://doi.org/10.48550/arXiv.1706.09516
33. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2018) CatBoost: unbiased boosting with categorical features. Adv Neural Inf Process Sys. https://doi.org/10.48550/arXiv.1706.09516
34. Esposito C, Landrum GA, Schneider N, Stiefl N, Riniker S (2021) GHOST: adjusting the decision threshold to handle imbalanced data in machine learning. J Chem Inf Model 61(6):2623–2640. https://doi.org/10.1021/acs.jcim.1c00160
35. Dahlin JL, Nissink JWM, Strasser JM, Francis S, Higgins L, Zhou H, Zhang Z, Walters MA (2015) PAINS in the assay: chemical mechanisms of assay interference and promiscuous enzymatic inhibition observed during a sulfhydryl-scavenging HTS. J Med Chem 58(5):2091–2113. https://doi.org/10.1021/jm5019093
36. Breiman L (2017) Classification and regression trees. Routledge, New York
37. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
38. Pedregosa F (2012) Scikit-learn: machine learning in python. Mach Learn. https://doi.org/10.48550/arXiv.1201.0490
39. XGBoost Documentation—xgboost 1.6.2 documentation. https://xgboost.readthedocs.io/en/stable/. Accessed 31 Aug 2022
40. Welcome to LightGBM's documentation!—LightGBM 3.3.2 documentation. https://lightgbm.readthedocs.io/en/v3.3.2/. Accessed 31 Aug 2022
41. Todeschini R, Consonni V (2000) Handbook of molecular descriptors. Methods Princ Med Chem. https://doi.org/10.1002/9783527613106
42. CatBoost - state-of-the-art open-source gradient boosting library with categorical features support. https://catboost.ai. Accessed 31 Aug 2022
43. Ustimenko A, Beliakov A, Prokhorenkova L (2022) Gradient boosting performs gaussian process inference. ArXiv. https://doi.org/10.48550/arXiv.2206.05608
44. Ustimenko, A.; Prokhorenkova, L. SGLB: Stochastic Gradient Langevin Boosting. http://arxiv.org/abs/2001.07248. Accessed 20 May 2022.
45. Sharchilev, B.; Ustinovsky, Y.; Serdyukov, P.; de Rijke, M. Finding Influential Training Samples for Gradient Boosted Decision Trees. arXiv March 12, 2018. http://arxiv.org/abs/1802.06640 Accessed 29 Jul 2022
46. Cortés-Ciriano I, Bender A (2019) Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. J Chem Inf Model 59(3):1269–1281. https://doi.org/10.1021/acs.jcim.8b00542
47. Fu G, Yi L, Pan J (2019) Tuning model parameters in class-imbalanced learning with precision-recall curve. Biom J 61(3):652–664. https://doi.org/10.1002/bimj.201800148
48. Feng Y, Zhou M, Tong X Imbalanced classification: a paradigm-based review. http://arxiv.org/abs/2002.04592. Accessed 10 Oct 2022
49. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64. https://doi.org/10.2307/2282330
50. RDKit. https://www.rdkit.org/. Accessed 09 May 2021
51. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD (2015) Hyperopt: a python library for model selection and hyperparameter optimization. Comput Sci Discov 8(1):014008. https://doi.org/10.1088/1749-4699/8/1/014008
52. Jiménez-Luna J, Grisoni F, Schneider G (2020) Drug discovery with explainable artificial intelligence. Nat Mach Intell 2(10):573–584. https://doi.org/10.1038/s42256-020-00236-4
53. Shapley L (1953) A value for n-person games. In: Kuhn HW, Tucker A (eds) Contributions to the theory of games (AM-28). Princeton University Press, Princeton
54. Sheridan RP (2019) Interpretation of QSAR models by coloring atoms according to changes in predicted activity: how robust is it? J Chem Inf Model 59(4):1324–1337. https://doi.org/10.1021/acs.jcim.8b00825
55. Hutter F, Hoos H, Leyton-Brown K (2014) An Efficient Approach for Assessing Hyperparameter Importance. In Proceedings of the 31st International Conference on International Conference on Machine Learning. ICML'14; JMLR.org: Beijing, China. 32:I-754–I-762. https://dl.acm.org/doi/10.5555/3044805.3044891
56. Durant JL, Leland BA, Henry DR, Nourse JG (2002) Reoptimization of MDL keys for use in drug discovery. J Chem Inf Model. https://doi.org/10.1021/ci010132r
57. Göller AH, Kuhnke L, Montanari F, Bonin A, Schneckener S, ter Laak A, Wichard J, Lobell M, Hillisch A (2020) Bayer's in silico ADMET platform: a journey of machine learning over the past two decades. Drug Discov Today 25(9):1702–1709. https://doi.org/10.1016/j.drudis.2020.07.001

Publisher's Note

C.

Paper 3 (chapter 5)

Accepted open access article in *ACS Central Science* 10, 4 (**2024)**.

by **Davide Boldini**, Lukas Friedrich, Daniel Kuhn and Stephan A. Sieber.

https://doi.org/10.1021/acscentsci.3c01517

ACS central science

http://pubs.acs.org/journal/acscii

Article

Machine Learning Assisted Hit Prioritization for High Throughput Screening in Drug Discovery

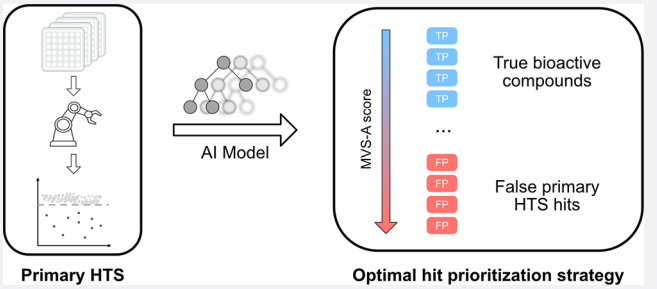
Davide Boldini, Lukas Friedrich, Daniel Kuhn, and Stephan A. Sieber*

Cite This: https://doi.org/10.1021/acscentsci.3c01517

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Efficient prioritization of bioactive compounds from high throughput screening campaigns is a fundamental challenge for accelerating drug development efforts. In this study, we present the first data-driven approach to simultaneously detect assay interferents and prioritize true bioactive compounds. By analyzing the learning dynamics during training of a gradient boosting model on noisy high throughput screening data using a novel formulation of sample influence, we are able to distinguish between compounds exhibiting the desired biological response and those producing assay artifacts. Therefore, our method enables false positive and true positive detection without relying on prior screens or assay interference mechanisms, making it applicable to any high throughput screening campaign. We demonstrate that our approach consistently excludes assay interferents with different mechanisms and prioritizes biologically relevant compounds more efficiently than all tested baselines, including a retrospective case study simulating its use in a real drug discovery campaign. Finally, our tool is extremely computationally efficient, requiring less than 30 s per assay on low-resource hardware. As such, our findings show that our method is an ideal addition to existing false positive detection tools and can be used to guide further pharmacological optimization after high throughput screening campaigns.



INTRODUCTION

High throughput screening (HTS) has significantly accelerated drug discovery efforts by allowing researchers to test large chemical libraries for bioactivity in a time and cost efficient manner, thus providing a crucial starting point for synthesizing small molecules with suitable pharmacological properties.[1−4]

However, one fundamental issue with HTS is its tendency to provide false positive readouts, either because the experimental response for a given hit compound is not reproducible or because it is not correlated with the intended biological activity.[5−9] The underlying causes for the false positive readout can be extremely heterogeneous, including colloidal aggregation,[10] autofluorescence,[11] interference with assay technology,[5,8] chemical reactivity,[12] metal impurities,[13] and measurement uncertainty.[14]

For these reasons, choosing which active compounds to prioritize for further pharmacological development after an HTS campaign still relies on further experimental profiling,[15−17] thus increasing the time and resources necessary to identify true hits and subsequently deliver a drug to the market.

This issue has garnered significant attention in the cheminformatics community, leading to the development of several in-silico tools for false positive detection in HTS data.[6,7,18−22] These methods are generally based on expert rule based approaches, for example Pan-Assay Interferent (PAINS) substructure filters,[5,8] or machine learning models trained on historical HTS data.[7,18,19] However, there are two main limitations to the use of these tools. First, they generally make assumptions concerning the assay interference mechanism, limiting their applicability to a narrow selection of false positives.[6,18] Furthermore, this aspect also limits their trustworthiness in identifying true positives since they can only prioritize compounds that are unlikely to be interferents according to that specific mechanism. For example, given an autofluorescence predictor for HTS interferent detection, even if it classifies a compound as nonfluorescent, that molecule might still be a false positive due to other phenomena, e.g., statistical fluctuations or colloidal aggregation. Second, these approaches depend on the chemical, biological, and technological space evaluated to generate them.[5,7] As such, their performance might be unreliable when evaluating compounds outside of the applicability domain of the model or when applied to HTS campaigns targeting unseen protein families, relying on new assay technologies and so forth.[7]

To speed up HTS hit triaging, we propose herein minimal variance sampling analysis (MVS-A), the first machine learning

Received: December 7, 2023
Revised: March 1, 2024
Accepted: March 1, 2024

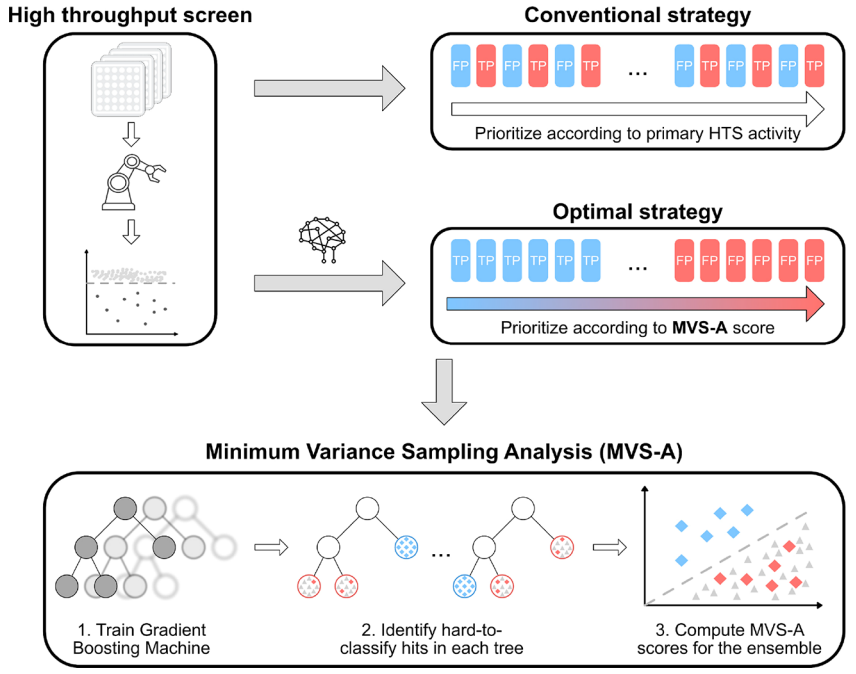


Figure 1. Illustration of our proposed approach. After an HTS campaign is carried out, the most active compounds in the primary screen are usually prioritized for further testing. However, this strategy often does not distinguish well between true positives (TP) and false positives (FP), leading to high false positive rates in the confirmatory screens. In our approach, we first fit a Gradient Boosting Machine classifier on the primary HTS screen data and compute MVS-A scores for each active compound. Problematic compounds according to the classifier will have high MVS-A scores and are likely false positives and vice versa for true hits. Selecting compounds according to their MVS-A score leads to reduced false positive rates in subsequent confirmatory screens and enables the identification of false positives in the primary HTS screen.

approach to simultaneously identify false positive compounds and prioritize true biologically active molecules in HTS data. Our approach is inspired by recent findings in gradient-based data valuation,[23−25] which showcase how tracing sample gradients during the training process can highlight mislabeled data in computer vision and natural language processing applications.[23,26] To make gradient-based data valuation more applicable out-of-the-box and reduce computational complexity, MVS-A is based on a novel formulation of sample influence for gradient boosting, thus enabling processing of large HTS data sets (e.g., above 300.000 compounds) in mere seconds. Because of this, MVS-A operates in an orthogonal fashion to prior false positive detection tools for HTS data: instead of requiring a preexisting library of assay interferents, it only requires training on the HTS itself, avoiding out-of-domain (OOD) applicability issues altogether. Additionally, since it does not make any assumptions about the interference mechanism, it can be used to successfully prioritize true positives.

To evaluate our approach, we curated a selection of 17 publicly available HTS data sets and 3 industrial ones with different sizes, class imbalance, biological targets, assay technology, and false positive rates. Our results show that MVS-A can outperform a variety of rule-based and data-driven baselines both at true positive and false positive identification.

RESULTS AND DISCUSSION

Using MVS-A to Prioritize Hits from HTS Campaigns. In recent years, analyzing sample gradient dynamics during supervised neural network training has attracted significant interest for modeling noisy data sets.[23−28] These methods enable quantification of the influence of each sample on the neural network weights once the model has been trained. When training on noisy data, such as HTS campaigns, it has been shown that sample influence correlates with the likelihood of being mislabeled, thus enabling the identification of both trustworthy and problematic samples. However, neural network based approaches are computationally expensive and sensitive to hyperparameters, especially for large, imbalanced molecular data sets such as HTS data,[29,30] making their use for nonexperts particularly challenging.

To tackle these limitations, we have developed minimum variance sampling analysis (MVS-A) to estimate sample influence in gradient boosting machines (GBM). GBM is a machine learning algorithm that fits an ensemble of decision trees in a sequence, each compensating for the mistakes of the previous tree. The advantages of using GBM instead of neural networks for computing sample influence are faster computation of importance scores, robust out-of-the-box performance, and classification performance on imbalanced HTS data, thus providing a good inductive bias for detecting false positive compounds.[31,32] In practice, the way MVS-A works is by quantifying how "unusual" a certain active compound is according to the GBM model when comparing it to the boundary it has learned to separate active and inactive molecules. If a compound is labeled as active in the training set, but the pattern learned by the GBM model contradicts that, it will have a high MVS-A score. Vice versa, if a bioactive molecule is easily identified as such by the classifier, it will have a low MVS-A score. These scores can be used accordingly to prioritize compounds for further testing, or a threshold can be set to label true positives and false positives depending on the hit validation budget. In this study, we consider for all data sets the bottom 10% of the hits as true positives and the top 10% as false positives, as done in another ranking evaluation study.[33]

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

As such, our proposed approach for ranking HTS hits goes as follows (Figure 1):

(1) We train a GBM classifier on the HTS data set of interest to distinguish hits from inactive compounds.
(2) We compute sample influence estimates for all hits via MVS-A.
(3) We sort all HTS hits according to their MVS-A score. False positives are likely to have high MVS-A scores and vice versa for true positives.

Thanks to its computational efficiency, this pipeline takes only a few seconds on low-end hardware, even for large HTS data sets. Crucially, our approach relies exclusively on the HTS data set of interest. As such, it does not rely on historical information on which compounds tend to be false positives for that assay technology (like, e.g., PAINS), nor on assumptions of which biophysical process is causing the interference (e.g., aggregation or autofluorescence predictors). This means that our method is inherently applicable to any assay technology and any region of the chemical space while being able to detect any type of interferent.

Finally, we provide a more in-depth discussion of the theory behind MVS-A in chapter 1 of the Supporting Information.

Constructing a Benchmark for HTS Hit Prioritization. To evaluate our proposed approach, we curated a selection of 17 data sets from publicly available HTS data,[34,35] for a total of 471370 unique compounds measured against 10 different protein families, using a variety of readout measurements and activity thresholds (Table 1, Table S1, Table S2, Table S3, and Table S4). We focused on HTS data sets where more than 200 hits were investigated both in primary and confirmatory screens, excluding campaigns where the false positive rate was above 95% or below 5%. Where possible, we prioritized the selection of assays targeting different protein families and confirmatory screen protocols.

Table 1. Summary Information for the Datasets Employed in This Study[a]

name	source	number of compounds	false positive %	number of hits
transporter	ref 33	306252	29%	2625
transcription	ref 33	344724	47%	2336
transcription_2	ref 33	301125	76%	2325
GPCR_2	ref 33	196068	79%	1980
GPCR_3	ref 33	63643	56%	2176
ion_channel	ref 33	305411	15%	2580
ion_channel_2	ref 33	104663	21%	4227
ion_channel_3	ref 33	305401	32%	1642
kinase	ref 34	321563	21%	234
GPCR	ref 34	325747	51%	5742
serine	ref 34	214071	91%	1262
transcription_3	ref 34	363477	81%	1790
ubiquitin	ref 34	330197	70%	1533
splicing	ref 34	293183	11%	2189
channel_atp	ref 34	343522	48%	1229
cysteine_protease	ref 34	344098	48%	1842
zinc_finger	ref 34	301590	48%	1132

[a]The number of hits defines the number of active compounds in the primary screen. The false positive percentage identifies the fraction of active compounds in the primary screen that were found to be inactive in the confirmatory screen.

As a result of this selection process, the false positive rates in our benchmark range from 11% to 91%, and the screened libraries evaluate different regions of the chemical space (Figure S1), thus covering a broad spectrum of HTS campaigns.

Each data set is generated from a primary screen, relying on single-dose measurements, and a confirmatory screen, which either adds replicates or assesses the dose−response activity against the same biological target. To define which molecules are considered bioactive in a given primary or confirmatory screen, we employed the original activity thresholds defined by the authors of the screening campaign. This ensures that our analysis accurately reflects real drug discovery campaigns as close as possible, where bioactivity criteria vary on a case-by-case basis, depending on the biological target and the purpose of the drug.

We define a compound as false positive if it was reported to be active in the primary screen but was found to be inactive or inconclusive in the confirmatory screen. Depending on the protocol employed for the confirmatory screen, different false positive types can be identified. When adding replicates, only errors associated with readout fluctuations or systematic errors (e.g., dust in the well plate) can be identified, while dose−response measurements enable detection of autofluorescence, colloidal aggregation, assay technology interference, and so forth.

Defining a Protocol to Assess HTS Hit Prioritization Strategies. For a given HTS data set, we run the MVS-A pipeline exclusively on the primary screening data. Then, we measure how effective our approach is at separating true actives and false positives by comparing its compound ranking to the confirmatory screening data. To evaluate the sorting performance, we measure top-K Precision, Enrichment Factor, and Boltzmann-Enhanced Discrimination of Receiver Operating Characteristic (BEDROC).[33] Since Precision is sensitive to the amount of noise in the data set, we scale it with respect to the false positive and true positive rate for each data set, making this metric more consistent across data sets. Therefore, a relative top-K precision score of 0.0 indicates that a given ranking is equal to random sorting, while values above 0.0 denote percent improvements over assay noise. We further discuss our metric selection in chapter 4 of the Supporting Information.

To contextualize the performance of MVS-A, we provide the following baselines:

- Detecting false positives according to REOS and GSK structural filters, two well-established rule-based approaches to detect false positives in HTS data.[36,37] We rank compounds in terms of the total number of flags according to both criteria.
- Prioritizing compounds for further screening according to activity in the primary HTS assay, the defacto approach for ranking hits both in academia and in the industry.[15] The underlying assumption here is that if a compound is very active in the primary screen, it is likely to have similar bioactivity in the confirmatory screen as well.
- Ranking according to CatBoost object importance,[38] another sample influence approach based on GBM relying on a different algorithm to compute importance scores. We discuss this method further in the Supporting Information.

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

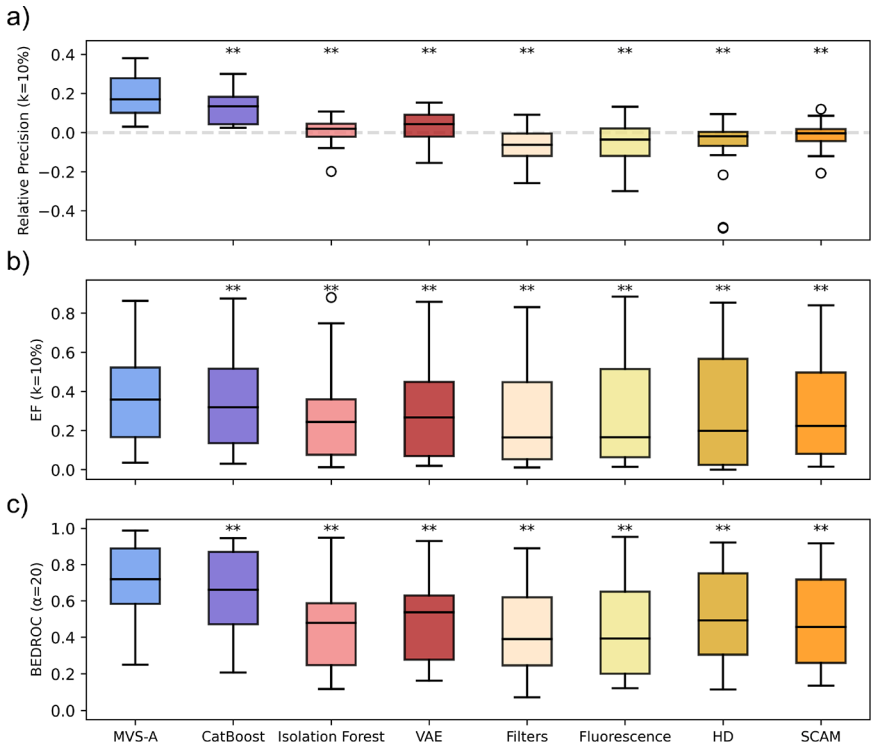


Figure 2. False positive detection performance across all data sets. Asterisks denote significance according to one-tailed Wilcoxon Signed Rank tests with Bonferroni correct (one asterisk corresponds to $\alpha = 0.05$, two asterisks to $\alpha = 0.01$). *P*-values are reported in Table S5. (a) Distribution of relative precision scores across all data sets. The dotted gray line denotes random performance. (b) Distribution of enrichment factor scores across all data sets. (c) Distribution of BEDROC scores across all data sets.

- Ranking according to Isolation Forest, a well-established anomaly detection algorithm based on decision tree ensembles.[39] We use the default parameters from the Scikit-Learn package.[40]
- Ranking according to the reconstruction error of a Variational Autoencoder (VAE), a popular deep learning approach for anomaly detection.[41−43] We implement a SMILES-based VAE using the architecture described by Gómez-Bombarelli et al.[44]

Additionally, to compare our approach with publicly available HTS interference predictors, we add the following baselines for false positive identification:

- Hit Dexter 3 (HD) for frequent hitter prediction.[6,18,45]
- SCAM Detective for colloidal aggregator identification.[46]
- An in-house autofluorescence predictor based on the models used by InterPred.[22] We discuss how we reproduced their featurization and optimization procedure in the Supporting Information.

MVS-A Achieves Best Performance in HTS False Positive Detection. In terms of false positive detection, MVS-A matches or outperforms, on average, all alternative methods across all metrics (Figure 2). The performance of our approach is mostly consistent across different metrics, meaning that MVS-A provides the best performance both when considering the top 10% predictions, as indicated by relative precision and enrichment factor, and when evaluating the entire ranking, as measured by BEDROC. Crucially, MVS-A outperforms all baselines across all metrics and data sets on 12 out of 17 data sets, while achieving second best performance on the remaining 5, making it an ideal option for out-of-the-box scenarios.

CatBoost object importance is the most competitive alternative; however, MVS-A still outperforms it on 16 out of 17 data sets across all metrics. Compared to this baseline, MVS-A provides an improvement of 29%, 6%, and 10% for relative precision, enrichment factor, and BEDROC respectively. Considering the differences in sample importance formulation between these methods, this result supports our method's assumption that focusing on the splitting decisions provides a better inductive bias for discovering mislabeled data.

Both anomaly detection methods, namely, Isolation Forest and VAE, struggle on this benchmark. Specifically, MVS-A outperforms them across all metrics on 16/17 and 15/17 data sets, respectively. Concerning VAE, this is likely due to the fact that these algorithms require large data sets (e.g., 10^6 compounds) to be trained properly,[44] while the number of hits per HTS is much lower. Regarding Isolation Forest, its performance is likely affected by the high dimensionality of the input molecular representations, rendering the use of random splits less effective.[39] In contrast, data valuation approaches like MVS-A object importance select the subset of informative features by first fitting a supervised classifier to distinguish active and inactive compounds, thus mitigating the issue of high dimensionality.

In comparison to GSK and REOS structural filters, MVS-A outperforms them on 16/17 data sets across all metrics. However, these alerts do not only detect false positives, but also focus on chemical moieties associated with target promiscuity or other undesirable pharmacological properties.[36,47,48] This mismatch then could explain the poor

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

performance observed in identifying false positives in this benchmark.

Finally, Hit Dexter, SCAM Detective, and the autofluorescence predictor show subpar false positive detection performance when compared to MVS-A, with our approach outperforming them across all metrics on 16/17, 17/17, and 15/17 data sets, respectively. This is likely because these approaches, unlike MVS-A, focus on specific interference mechanisms, while our benchmark makes no assumptions about the false positive origin. Furthermore, the performance of these baselines is likely degraded by applicability domain issues, while MVS-A is tailored to each specific screening campaign.

MVS-A Provides the Most Efficient HTS True Hit Prioritization Strategy. In line with the false positive retrieval benchmark, MVS-A on average matches or outperforms all other approaches across all metrics in terms of true hit detection (Figure 3) Specifically, it achieves the best

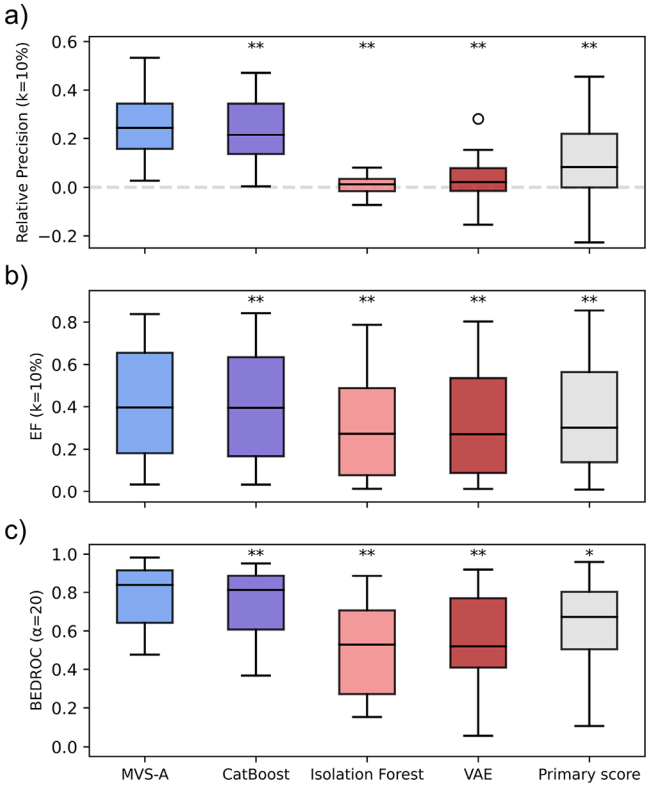


Figure 3. True positive detection performance across all data sets. Asterisks denote significance according to one-tailed Wilcoxon Signed Rank tests with Bonferroni correct (one asterisk corresponds to α = 0.05, two asterisks to α = 0.01). *P*-values are reported in Table S6. (a) Distribution of relative precision scores across all data sets. The dotted gray line denotes random performance. (b) Distribution of enrichment factor scores across all data sets. (c) Distribution of BEDROC scores across all data sets.

performance in 13 data sets out of 17 across all metrics, and it ranks second best in the remaining 4 data sets, further highlighting its potential as an optimal out-of-the-box solution.

On average, the most competitive baseline is again CatBoost object importance; however, MVS-A still outperforms it on 16/17 data sets. This further highlights that MVS-A is more effective at assessing sample influence than the previous state-of-the-art GBM algorithms since it detects high fidelity data more efficiently.

As for the false positive detection benchmark, anomaly detection methods provide subpar performance for true positive identification, with VAE showing slightly better performance than Isolation Forest. This is likely a consequence of the low data available for training the VAE and the high dimensionality of the input in the case of the Isolation Forest.

Finally, in comparison to the primary readout ranking, MVS-A outperforms it on 15 data sets, with average improvements of 50%, 13%, and 14% in terms of relative precision, enrichment factor, and BEDROC. This is especially impressive considering that ranking compounds according to their primary HTS readout is the industry standard for hit triaging in HTS campaigns. The relatively low performance of this method could be due to the fact that assay interferents can be outliers in terms of primary readout, for example, by exhibiting very strong autofluorescence, causing them to be at the top of the primary readout ranking. As such, this benchmark shows that our data-driven approach is more efficient at finding true actives than the currently used criteria for HTS hit triaging

MVS-A Identifies Structurally Diverse Interferents. While being able to correctly prioritize true positives and exclude false positives is a fundamental requirement for an HTS hit triaging strategy, retrieving a diverse set of compounds is also crucial. To assess this, we investigated the ability of our approach to identify heterogeneous true actives and assay interferents by measuring the fraction of unique Murcko scaffolds among the hits for both categories in each data set.

In terms of false positive variety, MVS-A selects the most diverse selection of interferents, peaking at around 95% scaffold diversity, closely followed by Hit Dexter and CatBoost object importance (Figure 4a). In general, data valuation algorithms such as MVS-A naturally tend to identify more varied interferents since they do not rely on the presence of specific molecular motifs in the false positives but rather highlight any active that deviates from the pattern they learned while training on the primary screening data. This more flexible definition of what constitutes a false positive then leads to the identification of more structurally different interferents, outperforming even anomaly detection algorithms. On the contrary, structural filters and assumption-based predictors are inherently biased toward specific chemical scaffolds, thus flagging more homogeneous compounds. One exception to this seems to be frequent hitters, which likely encompass several different interference mechanisms in their definition and, as such, have more diverse chemical structures.

This trend is inverted for true positive discovery, where both data valuation approaches tend to yield less diverse selections of true hits, centered around 60% scaffold diversity (Figure 4b). In this case, the true positives identified by MVS-A and CatBoost are the ones that fit well the learned class boundary between actives and inactives in the primary data. The boundary in this case tends to include only a limited region of the chemical space, leading to more structurally similar true actives. In contrast, primary readout ranking has no chemical bias in its selection criteria, thus retrieving the most diverse true positives. Finally, the scaffold diversity rate distribution across all data sets for the anomaly detection baselines is comparable with the one observed to randomly picking hits from each HTS (Table S4). This is because the true positives identified by these methods correspond to distribution inliers,

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

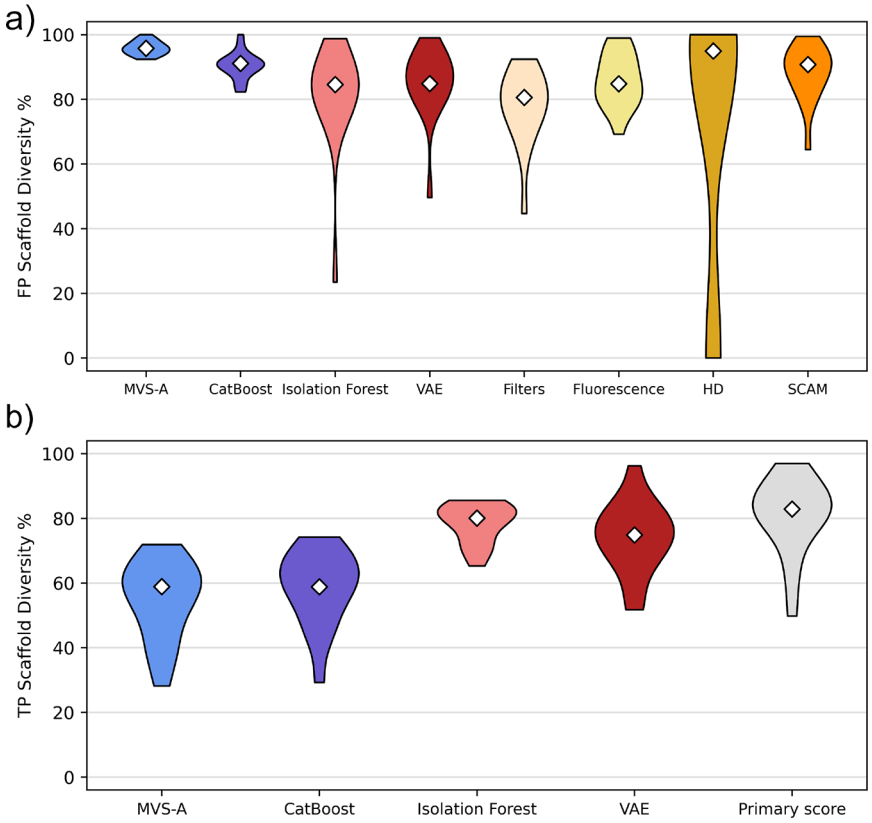


Figure 4. Structural diversity distribution analysis. White diamonds indicate the median of the distribution. (a) Distribution of the scaffold diversity scores across all data sets for false positive detection. (b) Distribution of the scaffold diversity scores across all data sets for true positive detection.

thus approximating the distribution of chemical motifs present in the training data.

MVS-A Identifies False Positives Belonging to Different Interferent Classes. By design, MVS-A makes no assumption concerning the interference mechanism of the false positive compounds in the primary screen; therefore, it should cover all types of interferents. To test this assumption, we measure across all data sets the fraction of compounds predicted to be false positives by our method that were also identified by the other assumption-based predictors (Figure S2).

MVS-A shows the highest overlap across all data sets with the autofluorescence predictor, with a median of 66%. This however is likely due to the nonselectivity of the autofluorescence predictor, which tends to flag the majority of compounds as fluorescent across all data sets (Table S7). These overconfident predictions could be due to applicability domain issues given that the training set used for this model originated from assays related to toxicological screening. Compared to the remaining in-silico predictors, MVS-A shows a median overlap of 51% with the colloidal aggregators identified by SCAM Detective, 33% with the structural filters from GSK and REOS and 8% with the frequent hitters detected by Hit Dexter.

Taken together, these results confirm the hypothesis that MVS-A can identify different classes of false positives while showing complementary performance with tools covering also compound promiscuity, such as frequent hitter predictors and general nuisance compound structural alerts.

Case Study I: Choline Transporter Inhibitor Screen from Vanderbilt University. To assess how well MVS-A would perform in a real drug discovery campaign, we investigated whether the true actives identified by our method are biased toward chemical moieties that would make them unsuitable for further pharmacological optimization. To do so, we re-evaluated the data set with the codename "transporter" from the publicly available HTS assays evaluated in this work. This assay was conducted in order to identify novel inhibitors for the presynaptic choline transporter (CHT), a potential therapeutic target for Alzheimer's disease and schizophrenia.[49] We chose this data set from our collection as a case study because the hits from its primary HTS screen were extensively validated by additional counterscreens and confirmatory assays (PubChem AID 488997). The goal of these experimental validation efforts was to identify potent selective CHT inhibitors eliciting the desired phenotypic response from primary HTS hits.

After 11 rounds of screening, only six compounds that were present in the primary HTS made it to the end of the pipeline, one of which, CHT4, was a false negative (Figure 5a). Crucially, all five true positives were immediately flagged as more promising than most other hits from the HTS campaign by MVS-A (Figure 5b), ranking within the top 20 primary HTS according to our method. Notably, among these five compounds there was also ML352, the best inhibitor from the screening campaign, which also showed suitable ADME properties.[49] In contrast, ranking by experimental readout from the primary screen is far less efficient, with ML352, CHT5, and CHT3 ranking between 150th and 500th, CHT1 around 750th, and CHT2 around 2300th (Figure 5b,d).

We then used MVS-A to rank primary inactive compounds in terms of false negative likelihood by sorting inactive

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

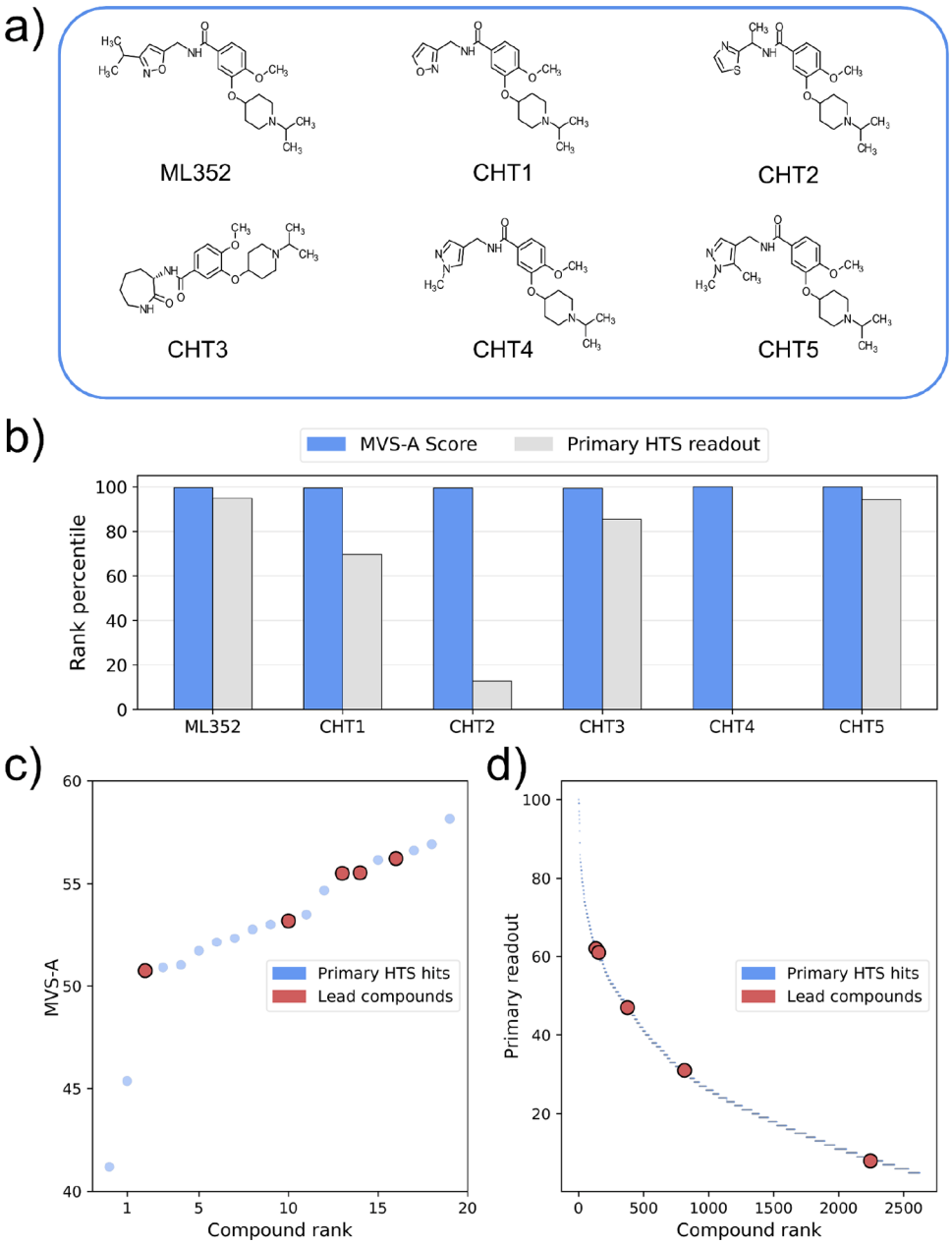


Figure 5. (a) Structures of the most relevant true positive compounds from the choline transporter inhibitor screening campaign. (b) Rank percentiles for the lead compounds according to MVS-A and the experimental readout from the primary HTS. (c) Compound ranking for the primary hits according to MVS-A. (d) Compound ranking according to the experimental readout for the primary hits.

compounds in terms of importance to the underlying GBM classifier according to our method. Crucially, our approach correctly identified CHT4 as the most likely false negative compound out of all primary inactives (Figure 5b). This finding is especially relevant, since mining dark chemical matter in HTS data is a promising but largely unexplored starting point for drug discovery,[50] given the lack of in-silico approaches to determine which samples to reinvestigate.

To summarize, in this case study MVS-A was able to identify the 6 most biologically relevant compounds just by observing the primary HTS data, including a false negative, while prioritizing molecules according to their experimental readout in the primary screen was a less efficient selection strategy. Additionally, this finding shows that MVS-A is not biased toward undesirable chemical moieties in terms of further pharmacological development.

Case Study II: Industrial HTS Campaigns from Merck KGaA. To further evaluate the applicability of MVS-A on real scenarios, we investigated three currently ongoing HTS campaigns from Merck KGaA, aimed at different biological targets (Table S8). Each of these data sets is larger than the largest publicly available data set we included in our study so far, thus providing a realistic benchmark for how our method would fare in industrial applications. Due to computational limitations, we could only test MVS-A, CatBoost and primary readout ranking on these data sets.

In terms of false positive detection, on average, MVS-A outperforms all baselines across all metrics (Table S9). Regarding true positive detection, on average, CatBoost and

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

MVS-A achieve similar performance, while primary readout ranking outperforms all alternatives in terms of precision and BEDROC (Table S10). In general, primary readout ranking performs much better as a baseline in these data sets, likely due to less assay noise compared to publicly available data, making the initial HTS screen more predictive of a compound's performance in further validation screens.

Limitations and Practical Guidelines for the Use of MVS-A. While MVS-A achieved excellent performance in terms of false positive and true positive detection, it still requires careful deployment for real use cases.

First, the performance of MVS-A can fluctuate from data set to data set, and it can be difficult to forecast how effective it will be for a given HTS data set. While we investigated the relationship between its performance and the HTS of interest, such as the protein target family, structural diversity of the data set (Figure S3), and the cross-validation performance of the GBM classifier (Figure S4), we could not detect meaningful correlation between these factors. As such, although MVS-A never performs worse than random picking in our benchmarks, the bias toward specific scaffolds in terms of true positive prioritization can be problematic if it is not associated with an improved true hit rate. This issue can be tackled, however, by hybrid hit selection strategies aimed at selecting diverse chemical scaffolds according to the MVS-A true hit likelihood.

Another factor that can influence the performance is the choice of molecular representation for the analysis. However, we observed only a 3% performance variation when using different molecular fingerprints and molecular descriptors (Figure S5), consistently with the results observed for molecular property prediction tasks.

In terms of computational cost, unlike other false positive predictors, MVS-A requires to be retrained for each new HTS data set. However, our testing shows that the algorithm is extremely efficient and lightweight, taking less than 5 s per data set on a server with an AMD Ryzen Threadripper 3970X 32-Core Processor and less than 30 s on a laptop with an AMD Ryzen 5 3600 6-Core Processor (Figure S6).

Finally, while MVS-A accurately distinguishes between interferents and true positives, it does not account for other relevant factors for hit prioritization such as promiscuity. As such, the ideal application of our approach is not as a stand-alone tool, but in conjunction with other in-silico tools, e.g., structural alerts or frequent hitter predictors, to get a global view of the pharmacological potential of each primary HTS hit. To highlight this, we revisited the top 20 ranked compounds from the CHT inhibitor screening campaign according to MVS-A, as discussed in Case Study I, focusing on the 15 compounds our approach incorrectly selected as the true hit. Six of those could be removed according to REOS and GSK filters, one according to Hit Dexter and one according to InterPred, while SCAM Detective flagged most compounds as potential colloidal aggregators (Table S10). As such, the synergistic combination of these approaches could have brought the true positive rate from 25% when using MVS-A on its own to 38%.

CONCLUSIONS

High throughput screening holds a key role in current drug discovery research, but its impact is limited by the presence of many false positive compounds, making further pharmacological development of bioactive compounds slower and more expensive. In this study, we introduced minimal variance sampling analysis, a novel approach inspired by data valuation methods to simultaneously prioritize true positive compounds and detect assay interferents in HTS data.

To test our proposed method, we have constructed a new benchmark consisting of 17 primary-confirmatory HTS data set pairs, encompassing a variety of biological targets, assay technologies, number of compounds, and false positive rates.

MVS-A consistently matches or outperforms the other baselines in terms of both false positive and true positive detection. Crucially, it provides average improvements up to 50%, 13%, and 14% in terms of relative precision, enrichment factor, and BEDROC against primary readout sorting, a popular heuristic used in the pharmaceutical industry for HTS hit prioritization. Concerning false positive discovery, our method can identify a wide range of structurally diverse interferents with low overlap with the predictions of prior in-silico tools focusing on compound promiscuity, making our method an excellent addition to HTS false positive detection pipelines.

Regarding hit prioritization, MVS-A was able to identify the most biologically relevant hits from a primary HTS campaign in a retrospective case study on publicly available data. Interestingly, one of the hits correctly detected by MVS-A was a false positive, highlighting the potential of our approach to detect promising compounds from dark chemical matter.

On the three data sets provided by Merck KGaA, MVS-A performs competitively in terms of false positive detection and false positive retrieval, indicating that our approach is also reliable in the chemical space typically explored in industrial screening campaigns.

Finally, our method is extremely computationally efficient, allowing processing of HTS data on a laptop in under 30 s with minimal RAM usage. In light of these results, we are confident MVS-A will help accelerate HTS hit triaging and will stimulate further research into data valuation approaches for handling large chemical data sets. We provide this tool as an open source package at https://github.com/dahvida/AIC_Finder.

ASSOCIATED CONTENT

Data Availability Statement

All PubChem assays investigated in this study can be accessed from PubChem according to their AID, as shown in Table 1. The Python environment, data sets, performance of each method across all metrics, data sets and replicates, and code required to reproduce the results are available at the following GitHub repository: https://github.com/dahvida/AIC_Finder.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acscentsci.3c01517.

Technical explanation of the theoretical aspects of MVS-A, the Methods section detailing how each approach was implemented, training time measurements, chemical space diversity analysis, false positive identification performance for the alternative machine-learning approaches, overlap analysis between the predictions of MVS-A and alternative machine-learning approaches (PDF)

AUTHOR INFORMATION

Corresponding Author

Stephan A. Sieber − *TUM School of Natural Sciences, Department of Bioscience, Center for Functional Protein*

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

Assemblies (CPA), Technical University of Munich, 85748 Garching bei München, Germany; orcid.org/0000-0002-9400-906X; Email: stephan.sieber@tum.de

Authors

Davide Boldini − *TUM School of Natural Sciences, Department of Bioscience, Center for Functional Protein Assemblies (CPA), Technical University of Munich, 85748 Garching bei München, Germany*

Lukas Friedrich − *The Healthcare business of Merck KGaA, 64293 Darmstadt, Germany*

Daniel Kuhn − *The Healthcare business of Merck KGaA, 64293 Darmstadt, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acscentsci.3c01517

Author Contributions

Conceptualization: D.B. Benchmarking: D.B. Software: D.B. Methodology: D.B. and L.F. Analysis of the results: all authors. Writing original draft: D.B. Writing review and editing: all authors. All authors have given approval to the final version of the manuscript.

Notes

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Merck KGaA Darmstadt for their generous support with the Merck Future Insight Prize 2020. This project was also cofunded by the European Union (ERC, breakingBAC, 101096911). D.B. thanks Maximilian Schuh and Joshua Hesse for their help revising the draft and creating the pictures.

■ REFERENCES

(1) Blay, V.; Tolani, B.; Ho, S. P.; Arkin, M. R. High-Throughput Screening: Today's Biochemical and Cell-Based Approaches. *Drug Discovery Today* **2020**, *25* (10), 1807−1821.

(2) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: New Data Content and Improved Web Interfaces. *Nucleic Acids Res.* **2021**, *49* (D1), D1388−D1395.

(3) Macarron, R.; Banks, M. N.; Bojanic, D.; Burns, D. J.; Cirovic, D. A.; Garyantes, T.; Green, D. V. S.; Hertzberg, R. P.; Janzen, W. P.; Paslay, J. W.; Schopfer, U.; Sittampalam, G. S. Impact of High-Throughput Screening in Biomedical Research. *Nat. Rev. Drug Discovery* **2011**, *10* (3), 188−195.

(4) Schneider, P.; Walters, W. P.; Plowright, A. T.; Sieroka, N.; Listgarten, J.; Goodnow, R. A.; Fisher, J.; Jansen, J. M.; Duca, J. S.; Rush, T. S.; Zentgraf, M.; Hill, J. E.; Krutoholow, E.; Kohler, M.; Blaney, J.; Funatsu, K.; Luebkemann, C.; Schneider, G. Rethinking Drug Design in the Artificial Intelligence Era. *Nat. Rev. Drug Discovery* **2020**, *19* (5), 353−364.

(5) Dahlin, J. L.; Nissink, J. W. M.; Strasser, J. M.; Francis, S.; Higgins, L.; Zhou, H.; Zhang, Z.; Walters, M. A. PAINS in the Assay: Chemical Mechanisms of Assay Interference and Promiscuous Enzymatic Inhibition Observed during a Sulfhydryl-Scavenging HTS. *J. Med. Chem.* **2015**, *58* (5), 2091−2113.

(6) Stork, C.; Chen, Y.; Šícho, M.; Kirchmair, J. Hit Dexter 2.0: Machine-Learning Models for the Prediction of Frequent Hitters. *J. Chem. Inf. Model.* **2019**, *59* (3), 1030−1043.

(7) David, L.; Walsh, J.; Sturm, N.; Feierberg, I.; Nissink, J. W. M.; Chen, H.; Bajorath, J.; Engkvist, O. Identification of Compounds That Interfere with High-Throughput Screening Assay Technologies. *ChemMedChem.* **2019**, *14* (20), 1795−1802.

(8) Baell, J. B.; Nissink, J. W. M. Seven Year Itch: Pan-Assay Interference Compounds (PAINS) in 2017—Utility and Limitations. *ACS Chem. Biol.* **2018**, *13* (1), 36−44.

(9) Sink, R.; Gobec, S.; Pečar, S.; Zega, A. False Positives in the Early Stages of Drug Discovery. *Curr. Med. Chem.* **2010**, *17* (34), 4231−4255.

(10) Auld, D. S.; Inglese, J.; Dahlin, J. L. Assay Interference by Aggregation. In *Assay Guidance Manual*; Markossian, S., Grossman, A., Brimacombe, K., Arkin, M., Auld, D., Austin, C., Baell, J., Chung, T. D. Y., Coussens, N. P., Dahlin, J. L., Devanarayan, V., Foley, T. L., Glicksman, M., Gorshkov, K., Haas, J. V., Hall, M. D., Hoare, S., Inglese, J., Iversen, P. W., Kales, S. C., Lal-Nag, M., Li, Z., McGee, J., McManus, O., Riss, T., Saradjian, P., Sittampalam, G. S., Tarselli, M., Trask, O. J., Wang, Y., Weidner, J. R., Wildey, M. J., Wilson, K., Xia, M., Xu, X., Eds.; Eli Lilly & Company and the National Center for Advancing Translational Sciences: Bethesda (MD), 2004.

(11) Hall, M. D.; Simeonov, A.; Davis, M. I. Avoiding Fluorescence Assay Interference—The Case for Diaphorase. *Assay Drug Dev. Technol.* **2016**, *14* (3), 175−179.

(12) Huth, J. R.; Mendoza, R.; Olejniczak, E. T.; Johnson, R. W.; Cothron, D. A.; Liu, Y.; Lerner, C. G.; Chen, J.; Hajduk, P. J. ALARM NMR: A Rapid and Robust Experimental Method To Detect Reactive False Positives in Biochemical Screens. *J. Am. Chem. Soc.* **2005**, *127* (1), 217−224.

(13) Hermann, J. C.; Chen, Y.; Wartchow, C.; Menke, J.; Gao, L.; Gleason, S. K.; Haynes, N.-E.; Scott, N.; Petersen, A.; Gabriel, S.; Vu, B.; George, K. M.; Narayanan, A.; Li, S. H.; Qian, H.; Beatini, N.; Niu, L.; Gan, Q.-F. Metal Impurities Cause False Positives in High-Throughput Screening Campaigns. *ACS Med. Chem. Lett.* **2013**, *4* (2), 197−200.

(14) Dragiev, P.; Nadon, R.; Makarenkov, V. Systematic Error Detection in Experimental High-Throughput Screening. *BMC Bioinformatics* **2011**, *12* (1), 25.

(15) Vincent, F.; Loria, P. M.; Weston, A. D.; Steppan, C. M.; Doyonnas, R.; Wang, Y.-M.; Rockwell, K. L.; Peakman, M.-C. Hit Triage and Validation in Phenotypic Screening: Considerations and Strategies. *Cell Chem. Biol.* **2020**, *27* (11), 1332−1346.

(16) Hughes, J.; Rees, S.; Kalindjian, S.; Philpott, K. Principles of Early Drug Discovery. *Br. J. Pharmacol.* **2011**, *162* (6), 1239−1249.

(17) Hevener, K. E.; Pesavento, R.; Ren, J.; Lee, H.; Ratia, K.; Johnson, M. E. Chapter Twelve - Hit-to-Lead: Hit Validation and Assessment. In *Methods in Enzymology*; Lesburg, C. A., Ed.; Modern Approaches in Drug Discovery; Academic Press, 2018; Vol. *610*, pp 265−309. DOI: 10.1016/bs.mie.2018.09.022.

(18) Stork, C.; Wagner, J.; Friedrich, N.-O.; de Bruyn Kops, C.; Šícho, M.; Kirchmair, J. Hit Dexter: A Machine-Learning Model for the Prediction of Frequent Hitters. *ChemMedChem.* **2018**, *13* (6), 564−571.

(19) Yang, Z.-Y.; Yang, Z.-J.; Dong, J.; Wang, L.-L.; Zhang, L.-X.; Ding, J.-J.; Ding, X.-Q.; Lu, A.-P.; Hou, T.-J.; Cao, D.-S. Structural Analysis and Identification of Colloidal Aggregators in Drug Discovery. *J. Chem. Inf. Model.* **2019**, *59* (9), 3714−3726.

(20) Irwin, J. J.; Duan, D.; Torosyan, H.; Doak, A. K.; Ziebart, K. T.; Sterling, T.; Tumanian, G.; Shoichet, B. K. An Aggregation Advisor for Ligand Discovery. *J. Med. Chem.* **2015**, *58* (17), 7076−7087.

(21) Lee, K.; Yang, A.; Lin, Y.-C.; Reker, D.; Bernardes, G. J. L.; Rodrigues, T. Combating Small-Molecule Aggregation with Machine Learning. *Cell Rep. Phys. Sci.* **2021**, *2* (9), No. 100573.

(22) Borrel, A.; Mansouri, K.; Nolte, S.; Saddler, T.; Conway, M.; Schmitt, C.; Kleinstreuer, N. C. InterPred: A Webtool to Predict Chemical Autofluorescence and Luminescence Interference. *Nucleic Acids Res.* **2020**, *48* (W1), W586−W590.

(23) Pruthi, G.; Liu, F.; Sundararajan, M.; Kale, S. Estimating Training Data Influence by Tracing Gradient Descent. *arXiv*,

November 14, 2020. DOI: 10.48550/arXiv.2002.08484 (accessed 2022-08-31).
(24) Feng, Y.; Tu, Y. Phases of Learning Dynamics in Artificial Neural Networks in the Absence or Presence of Mislabeled Data. *Mach. Learn. Sci. Technol.* **2021**, *2* (4), No. 043001.
(25) Pleiss, G.; Zhang, T.; Weinberger, K. Q.; Elenberg, E. Identifying Mislabeled Data Using the Area Under the Margin Ranking *arXiv*, 2020. DOI: 10.48550/arXiv.2001.10528.
(26) Akyurek, E.; Bolukbasi, T.; Liu, F.; Xiong, B.; Tenney, I.; Andreas, J.; Guu, K. Towards Tracing Knowledge in Language Models Back to the Training Data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*; Association for Computational Linguistics: Abu Dhabi, United Arab Emirates, 2022; pp 2429−2446.
(27) Lu, Y.; Bo, Y.; He, W. Noise Attention Learning: Enhancing Noise Robustness by Gradient Scaling. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 23164−23177.
(28) Toniato, A.; Schwaller, P.; Cardinale, A.; Geluykens, J.; Laino, T. Unassisted Noise Reduction of Chemical Reaction Datasets. *Nat. Mach. Intell.* **2021**, *3* (6), 485−494.
(29) Keshavarzi Arshadi, A.; Salem, M.; Firouzbakht, A.; Yuan, J. S. MolData, a Molecular Benchmark for Disease and Target Based Machine Learning. *J. Cheminformatics* **2022**, *14* (1), 10.
(30) Korkmaz, S. Deep Learning-Based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* **2020**, *60* (9), 4180−4190.
(31) Boldini, D.; Friedrich, L.; Kuhn, D.; Sieber, S. A. Tuning Gradient Boosting for Imbalanced Bioassay Modelling with Custom Loss Functions. *J. Cheminformatics* **2022**, *14* (1), 80.
(32) Jiang, D.; Wu, Z.; Hsieh, C.-Y.; Chen, G.; Liao, B.; Wang, Z.; Shen, C.; Cao, D.; Wu, J.; Hou, T. Could Graph Neural Networks Learn Better Molecular Representation for Drug Discovery? A Comparison Study of Descriptor-Based and Graph-Based Models. *J. Cheminformatics* **2021**, *13* (1), 12.
(33) Zhao, W.; Hevener, K. E.; White, S. W.; Lee, R. E.; Boyett, J. M. A Statistical Framework to Evaluate Virtual Screening. *BMC Bioinformatics* **2009**, *10* (1), 225.
(34) Butkiewicz, M.; Wang, Y.; Bryant, S. H.; Lowe, E. W., Jr.; Weaver, D. C.; Meiler, J. High-Throughput Screening Assay Datasets from the PubChem Database. *Chem. Inf. (Wilmington, Del)* **2017**, *3* (1), 1.
(35) Buterez, D.; Janet, J. P.; Kiddle, S. J.; Liò, P. MF-PCBA: Multifidelity High-Throughput Screening Benchmarks for Drug Discovery and Machine Learning. *J. Chem. Inf. Model.* **2023**, *63* (9), 2667−2678.
(36) Chakravorty, S. J.; Chan, J.; Greenwood, M. N.; Popa-Burke, I.; Remlinger, K. S.; Pickett, S. D.; Green, D. V. S.; Fillmore, M. C.; Dean, T. W.; Luengo, J. I.; Macarrón, R. Nuisance Compounds, PAINS Filters, and Dark Chemical Matter in the GSK HTS Collection. *SLAS Discovery* **2018**, *23* (6), 532−544.
(37) Walters, W. P.; Namchuk, M. Designing Screens: How to Make Your Hits a Hit. *Nat. Rev. Drug Discovery* **2003**, *2* (4), 259−266.
(38) Sharchilev, B.; Ustinovsky, Y.; Serdyukov, P.; de Rijke, M. Finding Influential Training Samples for Gradient Boosted Decision Trees. *arXiv*, March 12, 2018. http://arxiv.org/abs/1802.06640 (accessed 2022-07-29).
(39) Liu, F. T.; Ting, K. M.; Zhou, Z.-H. Isolation Forest. In 2008 *Eighth IEEE International Conference on Data Mining*; IEEE: Pisa, Italy, 2008; pp 413−422. DOI: 10.1109/ICDM.2008.17.
(40) Pedregosa, F.et al. Scikit-Learn: Machine Learning in Python. *J. Machine Learning Res.***2011***122825*
(41) Albuquerque Filho, J. E. D.; Brandao, L. C. P.; Fernandes, B. J. T.; Maciel, A. M. A. A Review of Neural Networks for Anomaly Detection. *IEEE Access* **2022**, *10*, 112342−112367.
(42) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* December 10, 2022. DOI: 10.48550/arXiv.1312.6114.
(43) Ruff, L.; Kauffmann, J. R.; Vandermeulen, R. A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T. G.; Muller, K.-R. A Unifying Review of Deep and Shallow Anomaly Detection. *Proc. IEEE* **2021**, *109* (5), 756−795.
(44) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268−276.
(45) Stork, C.; Mathai, N.; Kirchmair, J. Computational Prediction of Frequent Hitters in Target-Based and Cell-Based Assays. *Artif. Intell. Life Sci.* **2021**, *1*, No. 100007.
(46) Alves, V. M.; Capuzzi, S. J.; Braga, R. C.; Korn, D.; Hochuli, J. E.; Bowler, K. H.; Yasgar, A.; Rai, G.; Simeonov, A.; Muratov, E. N.; Zakharov, A. V.; Tropsha, A. SCAM Detective: Accurate Predictor of Small, Colloidally Aggregating Molecules. *J. Chem. Inf. Model.* **2020**, *60* (8), 4056−4063.
(47) Senger, M. R.; Fraga, C. A. M.; Dantas, R. F.; Silva, F. P. Filtering Promiscuous Compounds in Early Drug Discovery: Is It a Good Idea? *Drug Discovery Today* **2016**, *21* (6), 868−872.
(48) Dantas, R. F.; Evangelista, T. C. S.; Neves, B. J.; Senger, M. R.; Andrade, C. H.; Ferreira, S. B.; Silva-Junior, F. P. Dealing with Frequent Hitters in Drug Discovery: A Multidisciplinary View on the Issue of Filtering Compounds on Biological Screenings. *Expert Opin. Drug Discovery* **2019**, *14* (12), 1269−1282.
(49) Bollinger, S. R.; Engers, D. W.; Ennis, E. A.; Wright, J.; Locuson, C. W.; Lindsley, C. W.; Blakely, R. D.; Hopkins, C. R. Synthesis and Structure-Activity Relationships of a Series of 4-Methoxy-3-(Piperidin-4-Yl)Oxy Benzamides as Novel Inhibitors of the Presynaptic Choline Transporter. *Bioorg. Med. Chem. Lett.* **2015**, *25* (8), 1757−1760.
(50) Wassermann, A. M.; Lounkine, E.; Hoepfner, D.; Le Goff, G.; King, F. J; Studer, C.; Peltier, J. M; Grippo, M. L; Prindle, V.; Tao, J.; Schuffenhauer, A.; Wallace, I. M; Chen, S.; Krastel, P.; Cobos-Correa, A.; Parker, C. N; Davies, J. W; Glick, M. Dark chemical matter as a promising starting point for drug lead discovery. *Nat. Chem. Biol.* **2015**, *11*, 958−966.

https://doi.org/10.1021/acscentsci.3c01517
ACS Cent. Sci. XXXX, XXX, XXX−XXX

D.

Paper 4 (chapter 6)

Accepted open access article in *Journal of Cheminformatics* 16, 35 (**2024**).

by **Davide Boldini**, Davide Ballabio, Viviana Consonni, Roberto Todeschini, Francesca Grisoni and Stephan A. Sieber.

https://doi.org/10.1186/s13321-024-00830-3

Journal of Cheminformatics

RESEARCH

Open Access

Effectiveness of molecular fingerprints for exploring the chemical space of natural products

Davide Boldini[1*], Davide Ballabio[2], Viviana Consonni[2], Roberto Todeschini[2], Francesca Grisoni[3,4] and Stephan A. Sieber[1]

Abstract

Natural products are a diverse class of compounds with promising biological properties, such as high potency and excellent selectivity. However, they have different structural motifs than typical drug-like compounds, *e.g.*, a wider range of molecular weight, multiple stereocenters and higher fraction of sp^3-hybridized carbons. This makes the encoding of natural products via molecular fingerprints difficult, thus restricting their use in cheminformatics studies. To tackle this issue, we explored over 30 years of research to systematically evaluate which molecular fingerprint provides the best performance on the natural product chemical space. We considered 20 molecular fingerprints from four different sources, which we then benchmarked on over 100,000 unique natural products from the COCONUT (COlleCtion of Open Natural prodUcTs) and CMNPD (Comprehensive Marine Natural Products Database) databases. Our analysis focused on the correlation between different fingerprints and their classification performance on 12 bioactivity prediction datasets. Our results show that different encodings can provide fundamentally different views of the natural product chemical space, leading to substantial differences in pairwise similarity and performance. While Extended Connectivity Fingerprints are the de-facto option to encoding drug-like compounds, other fingerprints resulted to match or outperform them for bioactivity prediction of natural products. These results highlight the need to evaluate multiple fingerprinting algorithms for optimal performance and suggest new areas of research. Finally, we provide an open-source Python package for computing all molecular fingerprints considered in the study, as well as data and scripts necessary to reproduce the results, at https://github.com/dahvida/NP_Fingerprints.

Keywords Fingerprint, Natural products, Virtual screening, Similarity, Supervised classification

*Correspondence:
Davide Boldini
davide.boldini@tum.de
[1] TUM School of Natural Sciences, Department of Bioscience, Technical University of Munich, Center for Functional Protein Assemblies (CPA), 85748 Garching bei München, Germany
[2] Milano Chemometrics and QSAR Research Group, Department of Earth and Environmental Sciences, University of Milano-Bicocca, P.zza Della Scienza, 1, 20126 Milan, Italy
[3] Institute for Complex Molecular Systems and Department of Biomedical Engineering, Eindhoven University of Technology, Eindhoven, Netherlands
[4] Centre for Living Technologies, Alliance TU/e, WUR, UU, UMC Utrecht, Utrecht, Netherlands

Introduction

Natural products (NPs) are a source of inspiration for drug discovery due to their high potency and biological selectivity, which has translated in remarkable success in treating infectious diseases and cancer [1]. However, cheminformatic modeling of NPs has been limited because of their diversity from typical drug-like molecules (on which computational pipelines are usually developed), *e.g.*, in terms of their broader molecular weight distribution, multiple stereocenters, a higher fraction of *sp* [3]-hybridized carbons and extended ring systems [2, 3]. This issue is further compounded by a lack

of biological annotations for NPs [4] and the widespread presence of activity cliffs due to their highly specialized biological functions [1].

One of the key steps of cheminformatics pipelines is how to encode structural information into 'machine-readable' formats for further processing. This can be achieved through the so-called molecular descriptors [5], which convert selected molecular features into one or more numbers via a pre-defined algorithm. Among various descriptors applied to natural products [6, 7], molecular fingerprints—which convert a molecular structure into a vector—bear promise to capture structural information on natural products (*e.g.*, presence or absence of certain substructures). In fact, fingerprints generally provide satisfactory performance for quantitative structure–activity relationship (QSAR) modeling [8–10], even in the presence of activity cliffs [11]. Given the relevance of fingerprints in cheminformatics, over 30 years of research in the field have led to a broad and diverse selection of fingerprinting algorithms [12, 13]. However, while extensive research exists on the performance of these algorithms on synthetic, drug-like molecules, little is known about the best practices for natural products encoding.

Stemming from these observations, the aim of this study is to comprehensively compare and evaluate how different types of molecular fingerprints perform for modeling the NP chemical space, and ultimately to (a) provide effective recommendations to cheminformatics practitioners in the field of NPs, and (b) underscore future directions for the development of molecular fingerprints. We systematically compared 20 different molecular fingerprinting algorithms from four packages [14–18], on two cheminformatics tasks. First, we evaluate the similarity of fingerprints encoding using the COCONUT database [4], containing over 400,000 unique NPs from 52 different sources, and a wide variety of organisms, geographic locations and applications. Then, we evaluated the selected fingerprints for quantitative structure–activity relationship (QSAR) modeling, using 12 datasets from the CMNPD database. [19]

The diverse fingerprint behavior in similarity searches and QSAR modelling using NPs allowed us to shed on their effect in representing the chemical space of natural products.

Materials and methods

Dataset curation

Unsupervised analysis

We used the COCONUT database [4], which contains over 400,000 unique NPs from 52 different sources, including compounds from a wide variety of organisms, geographic locations and applications. We considered those natural products whose source organism was reported, as done in a previous study [20]. Solvent exclusion, salt removal and charge neutralization were performed with the ChEMBL structure curation package [21]. Compounds that failed this standardization step or have SMILES could not be parsed with RDKIT were removed. The resulting dataset included 129,869 unique natural products (Table 1), divided into six sources: plant, fungi, bacteria, marine, animal and mixed (defined for cases where the same natural product is produced by multiple organisms). Additional file 1: Table S1 details how many compounds were removed at each preprocessing step. Each class was characterized by a different diversity in terms of percentage of atomic scaffolds, which was computed by dividing the number of unique Bemis Murcko [22] scaffolds by the total number of compounds in each class (Table 1).

The distribution into classes (NP sources) is strongly skewed towards the plant class, encompassing 67.1% of total compounds, followed by fungi, bacteria, marine, mixed and animal (0.5%). In terms of compound diversity, there are four compounds per scaffold on average. The only outlier in this regard is the animal class, which has a much higher scaffold diversity rate (51.3%). This behavior might be related to the low number of NPs annotated for this class, or to the presence of acyclic natural products (e.g. linear peptides), making the Murcko scaffolds not as informative.

To compare the chemical space of NPs to typical drug-like compounds, we also included the Drug Repurposing Hub library in our analysis [23]. We preprocessed this dataset following the same procedure as for COCONUT, yielding 6776 unique drugs.

QSAR modeling

Concerning the supervised classification datasets, we standardized the natural products from the CMNPD database (Comprehensive Marine Natural Products Database) [19] as described above. We considered 12 different molecular property prediction tasks. To construct

Table 1 Summary of the data used in this study, collected and curated from COCONUT

Class	Number of compounds	Dataset %	Number of scaffolds	Scaffold diversity %
Plant	87,135	67.1	21,546	24.7
Fungi	15,516	11.9	4905	31.6
Bacteria	12,338	9.5	3824	31.0
Marine	8876	6.8	2443	27.5
Mixed	5290	4.1	1744	33.0
Animal	714	0.5	366	51.3
All	129,869	100	31,567	24.3

each task, we selected all NPs annotated with the desired property as the positive class and a random sample of NPs from CMNPD as the negative class, enforcing a minimum dataset size of 1000 compounds (Table 2).

Similar dataset generation procedures have been popularized for evaluating ligand-based virtual screening approaches [24–26], but they have the drawback of potentially introducing noise in the labels of the inactive compounds, since the negative class is constructed by sampling unlabeled molecules. However, this was necessary for our benchmark due to the scarcity of biological annotations for NPs, making it difficult to generate classification datasets where negative data had also been measured [3, 27].

Table 2 Summary of the classification datasets used in this study, collected and curated from CMNPD

Dataset	Number of compounds	Active compounds
Antibiotic	1000	112
Antiviral	1000	106
Antitumoral	1000	154
Antimalarial	1000	92
Antileishmanial	1000	20
Kinase C inhibition	1000	22
Serine Protease inhibition	1000	29
ATPase inhibition	1000	78
HIV	1000	178
Antifungal	1000	364
Anti-inflammatory	1000	156
Phosphatase inhibition	1000	95

Molecular fingerprints

In total, we analyzed 20 different fingerprinting algorithms belonging to five different categories (Table 3). We used the default calculation parameters provided by the source package for each fingerprint.

Five categories of fingerprints were considered, based on the type of molecular information they capture:

- *Path-based fingerprints* generate molecular features by analyzing the paths through the molecular graph given a pair of atoms and hashing them inside a fixed-size vector [16]. For example, Depth First Search (DFS) represents a compound by storing all unique paths in its graph, obtained by using each atom as the path starting point and moving away up to a number of bonds *d*. [32] Another example of this

Table 3 List of molecular fingerprints evaluated in this study, detailing for each the original publication year, the algorithm category, bit information type, number of bits, source package and parameters used for the calculation

Name	Year	Category	Type	Size	Source	Parameters
Topological Torsion (TT) [28]	1987	Path	Count	4096	RDKIT [14]	targetSize = 4
Atom Pair (AP) [29]	1985	Path	Count	4096	RDKIT [14]	N.A
Avalon [30]	2006	Path	Count	1024	RDKIT [14]	N.A
Daylight [31]	1973	Path	Binary	1024	CDK [15]	Depth = 7
Depth First Search (DFS) [32]	2005	Path	Binary	4096	jCompoundMapper [16]	Depth = 7
All Shortest Paths (ASP) [16]	2011	Path	Binary	4096	jCompoundMapper [16]	Depth = 7
RDKIT [14]	2012	Path	Binary	2048	RDKIT [14]	Depth = 7
Pharmacophore Pairs (PH2) [33]	2006	Pharmacophore	Binary	4096	jCompoundMapper [16]	N.A
Pharmacophore Triplets (PH3) [33]	2006	Pharmacophore	Binary	4096	jCompoundMapper [16]	N.A
MACCS [34]	2002	Substructure	Binary	166	RDKIT [14]	N.A
PubChem [35]	2009	Substructure	Binary	881	CDK [15]	N.A
ESTATE [36]	1995	Substructure	Binary	79	CDK [15]	N.A
Klekota-Roth (KR) [37]	2008	Substructure	Binary	4860	CDK [15]	N.A
Extended Connectivity (ECFP) [38]	2010	Circular	Binary	1024	RDKIT [14]	Radius = 2
Functional Class (FCFP) [38]	2010	Circular	Binary	1024	RDKIT [14]	Radius = 2
RAD2D [39]	2004	Circular	Binary	4096	jCompoundMapper [16]	N.A
LSTAR [16]	2011	Circular	Binary	4096	jCompoundMapper [16]	N.A
LINGO [40]	2005	String	Binary	1024	CDK [15]	N.A
MinHashed (MHFP) [18]	2018	String	Categorical	1024	Ref. [19]	Radius = 3
MinHashed Atom Pair (MAP4)17	2020	String	Categorical	1024	Ref. [18]	Radius = 2

class of algorithms are Atom Pair fingerprints (AP), where a molecule is described by collecting all possible triplets of two atoms and the shortest path connecting them [29].

- *Pharmacopohore fingerprints*, which are a variation of path-based fingerprints, where atoms are described by whether they are a pharmacophore point (e.g. whether they are hydrogen bond donors or acceptors) [33]. This leads to bit vectors that are less related to the compound structure, but instead try to encode how the molecule interacts with its chemical environment. Examples of this class of algorithms are Pharmacophore Pairs (PH2) and Pharmacophore Triplets (PH3) [33].
- *Substructure-based fingerprints,* in which each bit encodes whether the compound contains a predefined structural moiety [34, 37]. Examples of this class of algorithms are the MACCS structural keys and the PUBCHEM fingerprints [34, 35].
- *Circular fingerprints* also break up a target compound into different fragments like substructure-based fingerprints, but instead of relying on expert-defined structural patterns, they construct them dynamically from the molecular graph for each compound [38, 39]. To do so, they initially represent each atom according to some properties, such as atomic mass or valence. Then, for each atom, the numerical identifier of neighboring atoms is added, thus generating a fragment identifier. This process can be repeated several times, progressively increasing the radius of the neighborhood to consider when aggregating information. Finally, all unique fragments for a given molecule are hashed into a fixed-size vector. Typically, the difference between fingerprints belonging to this class lies in using different properties for the atom identifiers. For example, Extended Connectivity fingerprints (ECFP) use features such as the atomic number, atomic charge and so forth, while Functional Class fingerprints (FCFP) consider whether the atom is basic, acid, a hydrogen bond donor/acceptor etc [38].
- *String-based fingerprints* generate molecular representations by operating on the SMILES string of the compound, instead of its graph representation [18, 40]. For example, for a given dataset, LINGO fingerprints fragment the SMILES strings in fixed-size substrings and compute the total number of unique substrings across all compounds [40]. Then, each compound is encoded according to which SMILES substrings in the set it contains, using either counts or binary values. Another example of string-based algorithms are the MinHashed fingerprints (MHFP) [18]. This method works similarly to circular fingerprints, but instead of using atom identifiers, it considers the SMILES substring of a given fragment as its identifier. Each fragment identifier is then stored in a fixed-size vector via MinHash. MinHashed Atom Pair fingerprints (MAP4) [17] work similarly, but also consider the topological distance between atom pairs in the fragment for generating the fragment identifier.

Molecular fingerprints can be further characterized according to the information they encode in each element of the vector: binary fingerprints indicate the presence or absence of a given molecular pattern, count-based fingerprints have integer values specifying the number of occurrences of a given fragment and categorical fingerprints use numerical identifiers to describe the chemical motifs in the compound. [15–18]

Similarity metrics

We used the Jaccard-Tanimoto similarity [41] to assess pairwise similarities between compounds for all fingerprints. For categorical fingerprints (MAP4 and MHFP), we used a modified version of the Jaccard-Tanimoto similarity which considers two bits as a match if they contain exactly the same integer, as introduced in a previous study [17, 18, 20]. To ensure comparability, count-based fingerprints were converted into binary bits, by only encoding whether a fragment is present or absent, and then pairwise similarities were measured as for the other encodings. This ensures that any variation in pairwise similarities between two fingerprint types is exclusively related to differences in how the vectors are computed, and not due to using different metrics.

Pairwise distribution correlation analysis

For each type of fingerprint, evaluating all pairwise similarities on all compounds from the preprocessed version of the COCONUT dataset would be computationally infeasible, given that this would require calculating more than 8 billion similarity values. To mitigate this, we adopted a repeated resampling procedure which considered batches of 10,000 randomly selected NPs to compute the similarity, as:

- Given a sample of $n = 10,000$ compounds, we computed their fingerprints according to the 20 considered algorithms (Table 1), and for each type of fingerprint all the corresponding pairwise similarities.
- We concatenated the pairwise similarities in a matrix $\mathbf{B}(m \times p)$, with $m = \frac{10000*9999}{2} = 49995000$ and $p = 20$, and calculated mean, standard deviation, median and percentiles of the distribution of the

compound pairwise similarities for each type of fingerprint.
- Then, we computed the correlation matrix of **B**, yielding a matrix **C**(20 × 20), which describes how well each fingerprint correlates with one another in terms of pairwise similarities for a given natural product batch.
- Finally, once all batches were processed, we averaged all statistics across all 50 iterations.

The same procedure was repeated for the Drug Repurposing Hub dataset, but since it only has 6776 unique compounds, the procedure was carried out without the use of batches.

Unsupervised embeddings

We computed Uniform Manifold Approximation and Projection (UMAP) [42] embeddings for each fingerprint, using different metrics for each fingerprint numerical type as described in the Similarity metrics section. Each other parameter was set to its default value from the UMAP package [43]. We focused our analysis on the first batch of 10,000 molecules we used for the pairwise correlation analysis, since using the entire dataset would have been computationally infeasible. We verified that the class distribution and the chemical diversity for each batch is consistent with the values obtained for the whole dataset (Additional file 1: Tables S1-S2), ensuring that the UMAP analysis of the batch is representative of the entire chemical space we investigated.

Classification

To assess how well each fingerprint can be used for QSAR modeling of natural products, we evaluated them on 12 different bioactivity prediction datasets. Each classification dataset (Table 2) was divided in three folds using an 80:10:10 ratio between training, validation and test set with scaffold split [44]. For each fingerprint type, we then trained two models:

- *Random Forest classifier (RF)* [45]. Bayesian hyperparameter optimization for 20 iterations, training on the training split and measuring the ROC-AUC on the validation set (hyperparameters: number of trees between 50 and 500 with a step of 50, maximum tree depth between 5 and 12 with a step of 2, the minimum number of samples per split between 2 and 20, minimum number of samples per leaf between 2 and 100, number of features as a choice between the logarithm, the square root or 10% of the fingerprint size). We finally trained on the training set and evaluated the performance on the test set with 5 replicates.
- *Dense Neural Network (DNN)* [46] with 2 hidden layers, batch normalization and dropout. Each DNN was trained for 100 epochs using AdamW as the optimizer and binary cross-entropy as the loss function on the training set. The parameters were optimized via Bayesian optimization for 20 iterations according to the ROC-AUC on the validation set. We tuned the number of units per layer (between 128 and 512 with a step of 128), the dropout rate (between 0 and 0.4), the learning rate (between 0.0001 and 0.05) and the batch size (between 16 and 64 with a step of 8). Once the optimal hyperparameters were determined on the validation set, we retrained on the training set and measured all metrics on the test set, repeating the procedure 5 times.

The classification performance was quantified using precision, recall, specificity, Matthews Correlation Coefficient (MCC), F1 score, balanced accuracy, ROC-AUC and PR-AUC [47]. Our selection ensures that our evaluation encompasses all aspects of a given classifier's performance and is robust to class imbalance [48, 49]. To assess whether the any fingerprint was ranked differently than the others across all datasets, we first performed a Friedman test for each classification metric and classification model [50]. If the outcome of the Friedman test was statistically significant ($\alpha < 0.05$), we then performed post-hoc tests (2-tailed Wilcoxon signed rank test with Benjamini–Hochberg correction, $\alpha < 0.05$) to identify which fingerprint pair was significantly different [51, 52].

Hardware and software

The analysis and calculation pipelines were implemented in Python 3.8, using JPype 1.4.1 to access packages originally written in Java. We used RDKIT 2022.9.5, CDK 2.2 and jCompoundMapper 1.0 for computing fingerprints, scipy 1.8.1 and numpy 1.22.3 for computing Tanimoto similarity and performing statistical tests, statsmodels 0.15 for adjust p-values with the Benjamini–Hochberg correction [53], RDKIT 2022.9.5 and chembl_structure_pipeline 1.2.0 for compound standardization, hyperopt 0.2.7 for Bayesian hyperparameter optimization [54], Pytorch 2.1.0 [55] for training the DNN models and scikit-learn 1.2.2 [56] for training the RF models and computing classification metrics. All calculations were carried out on a server with an AMD Ryzen Threadripper 3970 × 32-core CPU and 128GB of RAM, using all threads available. The code for reproducing the results, calculating all the considered fingerprints, along with the performance metrics for each individual dataset and classifier are provided for free in the following Github repository: https://github.com/dahvida/NP_Fingerprints.

Results and discussion

Pairwise similarity distribution

We first analyzed the distribution of pairwise similarities across the COCONUT dataset (Fig. 1 and Table 4) and the Drug Repurposing Hub compounds (Additional file 1: Figure S1) to understand which fingerprints provide a more granular view for NPs and whether these patterns differ with drug-like molecules.

On the COCONUT dataset, Pharmacological fingerprints (PH2 and PH3) have the broadest distribution of pairwise similarities as well as the highest median Jaccard-Tanimoto similarity. Crucially, both distributions consistently reach similarity scores above 0.95, especially for PH2, indicating that even though the dataset is without replicates, according to these embedding many compounds are nearly indistinguishable. This is consistent with how this class of fingerprint is computed: instead of capturing information pertaining to the molecular structure, these embeddings try to describe molecules in terms of how they interact with their biological environment through their pharmacophores. As such, compounds that have very different chemical structures can still have identical pharmacophoric points, which is reflected by their high similarity scores in terms of PH2 and PH3 fingerprints. This shows that these featurization approaches are well suited for scaffold hopping in the NP chemical space, but their inability to separate structurally different compounds might be problematic for other QSAR applications. On the Drug Repurposing Hub both fingerprints achieve significantly lower median Jaccard-Tanimoto similarities (Mann Whitney test with Benjamini–Hochberg correction, $\alpha = 0.05$), especially PH3. This might be due to the smaller dataset size and higher scaffold diversity compared to COCONUT (62% instead of 24%), which generally lowers all median Jaccard-Tanimoto similarities for all fingerprints.

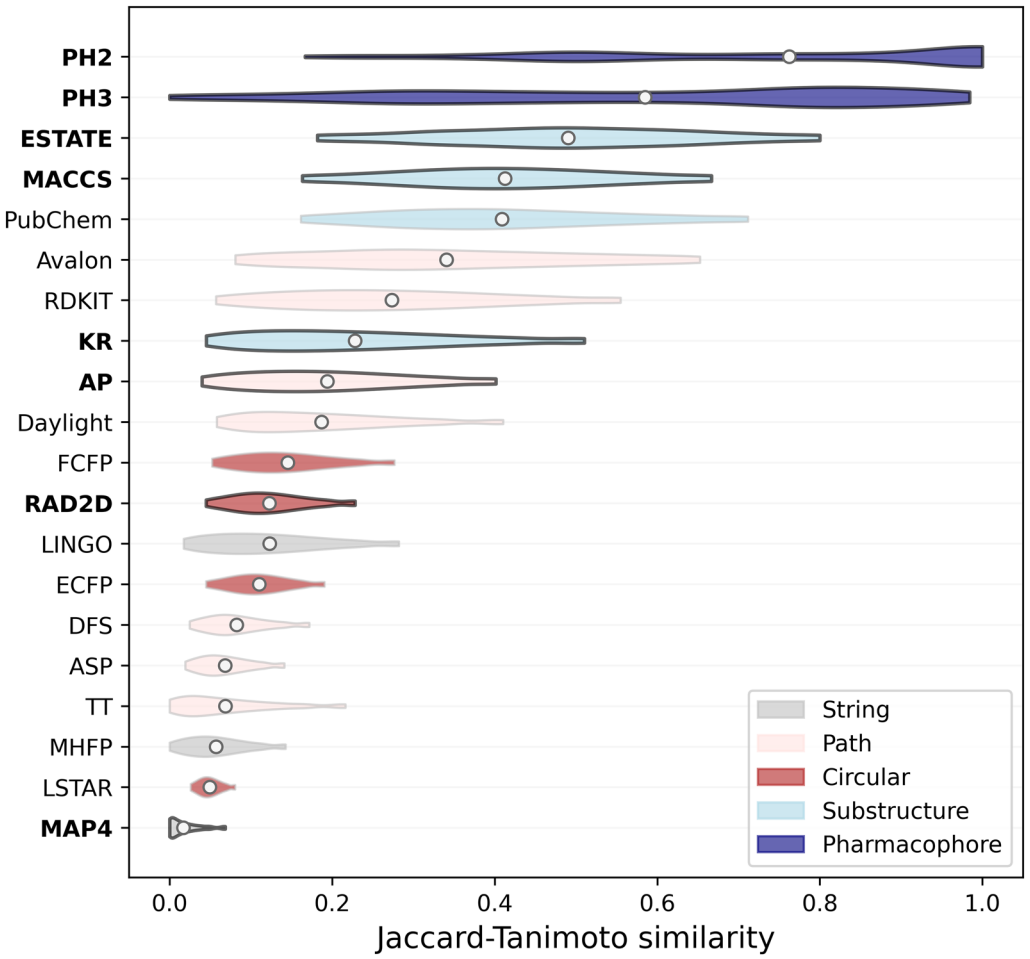


Fig. 1 Jaccard-Tanimoto similarity distribution for each fingerprint across all possible pairwise comparisons in the natural product dataset. Violin plots indicate the percentiles of the distribution of Jaccard-Tanimoto similarities, with the circle indicating the median similarity value. The fingerprints where the similarity distribution on natural products is significantly different than the one obtained for drug-like compounds are highlighted in bold (Mann Whitney tests with Benjamini–Hochberg correction, $\alpha = 0.05$)

Table 4 Distribution statistics for the pairwise Jaccard-Tanimoto similarity scores obtained by each fingerprint across all batches of the COCONUT dataset

Fingerprint	Minimum	25th percentile	50th percentile	75th percentile	Maximum
MAP4	0.000	0.002	0.011	0.026	0.067
LSTAR	0.026	0.039	0.048	0.059	0.080
MHFP	0.000	0.028	0.052	0.082	0.141
TT	0.000	0.023	0.055	0.103	0.212
ASP	0.020	0.043	0.064	0.090	0.140
DFS	0.026	0.054	0.077	0.107	0.169
ECFP	0.046	0.082	0.108	0.137	0.190
LINGO	0.018	0.065	0.114	0.173	0.279
RAD2D	0.047	0.087	0.118	0.154	0.226
FCFP	0.053	0.099	0.139	0.186	0.275
Daylight	0.059	0.111	0.171	0.249	0.404
AP	0.042	0.113	0.184	0.267	0.399
KR	0.047	0.125	0.210	0.317	0.504
RDKIT	0.062	0.166	0.261	0.371	0.550
Avalon	0.084	0.211	0.326	0.467	0.648
PubChem	0.167	0.294	0.396	0.516	0.706
MACCS	0.168	0.313	0.410	0.511	0.667
ESTATE	0.186	0.364	0.500	0.615	0.799
PH3	0.036	0.322	0.638	0.830	0.952
PH2	0.228	0.500	0.875	1.000	1.000

Another factor could be a larger range of pharmacophoric arrangements between the drugs considered for the analysis, consistently with the broad range of therapeutic targets of the molecules of this library. In that case, this pattern would affect PH3 more since it considers triplets instead of pairs, which leads to a higher number of potential combinations.

Next, substructure-based fingerprints like MACCS, ESTATE, PubChem and KR tend to achieve the highest Jaccard-Tanimoto similarity scores. This is consistent with their reliance on predefined fragments, rather than processing each molecular graph individually. Since the fragments chosen by these fingerprints were defined for small molecules, only a fraction of them is usually found in NPs, while other highly informative NP-like substructures are not encoded. This reduces the average bit variance across the fingerprints, leading to more similar vectors overall. These types of embeddings can therefore be problematic for the NP chemical space, unless custom fragments are added to account for the molecular distribution shift and feature selection is used to remove uninformative bits. This issue seems especially pronounced for MACCS and KR, since they achieve significantly lower median similarity scores (Mann Whitney test with Benjamini–Hochberg correction, $\alpha = 0.05$) on the Drug Repurposing Hub, shifting from 0.40 and 0.21 to 0.32 and 0.13. In contrast, PubChem and ESTATE remain comparable. This trend reflects the focus MACCS and KR have on drug discovery, thus biasing the fragment choice on relevant motifs for the drug-like chemical space. [34, 37]

Both path-based and circular fingerprints have median values of Jaccard-Tanimoto similarity around 0.1, and narrower score distributions. Two exceptions to this pattern are RDKIT, which has a comparable distribution to substructure-based encodings, and LSTAR, which has a very narrow distribution with a lower median similarity than other circular or path-based fingerprints. A similar trend is observed on the Drug Repurposing Hub, with path-based and circular fingerprints being distributed between 0.2 and 0.1 median Jaccard-Tanimoto similarity scores.

When it comes to MinHashed fingerprints, the low median Jaccard-Tanimoto scores obtained by MAP4 on both COCONUT and the Drug Repurposing Hub (less than 0.02) could be related to two factors. First, this fingerprint uses categorical encodings, which means that their similarity is computed via the modified Jaccard-Tanimoto similarity. According to that metric, for two bits to be considered a match it is not enough that they are both non-zero, but they must have the same integer value. As such, the fraction of matching bits given two fingerprints of this type tends to be much lower compared to binary fingerprints.

Second, it could be that MinHashing paths rather than circular fragments lead to more potential categorical values for each bit, reducing the number of bit matches when comparing two fingerprints. This would explain why MHFP has higher median pairwise Jaccard-Tanimoto similarity.

To further analyze the distribution of pairwise similarity scores, we evaluated the average "bit saturation" [57] of each fingerprint on the COCONUT and Drug Repurposing Hub datasets (Additional file 1: Table S5). On average, most fingerprints have higher saturation scores for natural products than for synthetic drugs, indicating the presence of larger, and more complex molecular structures [1]. One exception to this trend is substructure fingerprints, which have lower bit saturation on natural products than drug-like compounds. This is caused by the presence of uninformative fragments for natural products in the fingerprint definition, leading to less bits being set when encoding a given compound.

Fingerprint correlation analysis

To better evaluate which fingerprints provide different views of the NP chemical space, we calculated the Pearson correlation coefficient between each pairwise similarity score across all fingerprints (Fig. 2a). It is immediately apparent that both pharmacological fingerprints (PH2 and PH3) are outliers, given that they are extremely correlated between each other and almost completely uncorrelated with all others. This could be related to the fact that, unlike the other fingerprints analyzed, these fingerprints describe the occurrence of 'fuzzy' pharmacophoric points, rather than focusing on the presence or occurrence of functional groups and substructures.

When evaluating the correlations between the other fingerprints, it becomes clear that some fingerprints are highly correlated (above 0.8) with each other. MAP4 and MHFP (string fingerprints), as well as DFS and ASP (pharmacophore fingerprints) show high Pearson correlation coefficients (0.85 and 0.92 respectively). This is consistent with the fact that they belong to the same class, and hence are based on a similar featurization

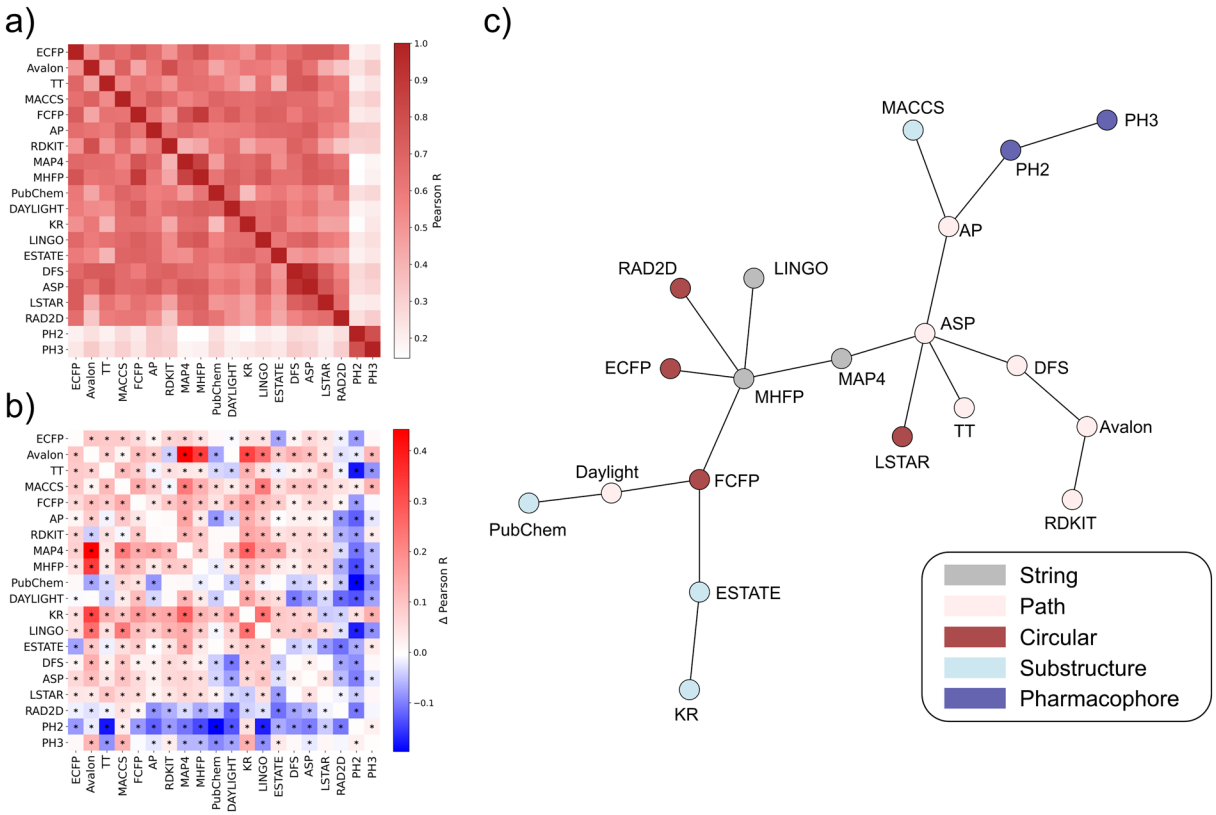


Fig. 2 Jaccard-Tanimoto similarity correlation analysis for all fingerprints. **a** Correlation matrix for all fingerprints evaluated in this study on the COCONUT dataset. **b** Difference between the correlation matrix obtained for the COCONUT dataset and for the Drug Repurposing Hub. Positive values indicate higher fingerprint correlation in the NP space, while negative values denote higher correlation in the drug-like space. Asterisks denote statistical significance according to one-sample Mann Whitney tests with Benjamini–Hochberg correction ($\alpha = 0.05$). **c** MST constructed from the fingerprint correlation matrix obtained for the NP chemical space. Each encoding is colored on the basis of its category

strategy. The first pair is especially interesting, given that while they both rely on SMILES substrings, MAP4 relies on topological distances between atom pairs, while MHFP considers circular neighborhoods around atoms for its fragments. This difference is also consistent when looking at their correlation with other circular fingerprints, such as ECFP and FCFP: MHFP strongly correlates with both (0.77 and 0.88), while MAP4 to a lesser extent (0.67 and 0.77).

To quantitatively assess which fingerprint correlation pairs change the most when considering the NP chemical space specifically, we first computed the correlation matrix for the Drug Repurposing Hub dataset (Additional file 1: Figure S2) and then calculated the Pearson R difference between the values obtained for NPs and the ones for drugs (Fig. 2b). For most encoding pairs, the difference is statistically significant, as shown in Fig. 2b (one-sample Mann Whitney tests with Benjamini Hochberg correction, $\alpha = 0.05$). Most fingerprints are more correlated in the NP space than in the drug-like space, with an average Pearson R difference of around 0.1, except for PH2 and PH3, which instead are less correlated to the others. The correlation increase for the majority of fingerprints likely reflects the fact that many bits are less informative for NPs than they are for drugs, thus reducing the ability of different fingerprints to capture molecular similarity from different perspectives. Notably, the correlation difference between Avalon and KR, MAP4 and MHFP is especially high (0.4), indicating that their chemical space mapping is very similar with NPs but not with drug-like compounds. On the other hand, the correlation decrease observed for PH2 and PH3 hints at the fact that similarities computed using these encodings tend to be outliers in the NP chemical space, as observed when evaluating their distribution and as discussed below when analyzing their unsupervised embeddings.

Another key difference between natural products and drug-like compounds is that the former tend to have a higher number of repetitive chemical moieties, which can be accurately captured by using count-based fingerprints. To evaluate how using counts affects the encoding of natural products, we repeated the Pearson correlation analysis for all count-based fingerprints (AP, TT and Avalon) for both COCONUT and Drug Repurposing Hub datasets (Additional file 1: Table S6). While there is a consistently high similarity score correlation between using counts and binary bits for a given fingerprint (e.g. AP has a Pearson R of 0.75 on the COCONUT dataset), there is a statistically significant difference for all fingerprints in how correlated counts and bits are when comparing natural products and drug-like compounds. Specifically, AP and Avalon show less correlation on natural products than on drug-like molecules, decreasing by 0.01 and 0.03 in terms of Pearson R respectively. In contrast, TT shows higher Pearson R on medicinal chemistry compounds. These results suggest therefore that count-based AP and Avalon fingerprints are more appropriate at capturing repetitive chemical moieties found in natural products, since there is larger disagreement between counts and binary fingerprints in terms of molecular similarity.

Visualizing fingerprint similarity via minimum spanning tree

To further aid in the visualization of the similarities between fingerprints, we constructed a Minimum Spanning Tree (MST) [58] from the correlation matrix (Fig. 2c). The Minimum Spanning Tree was performed by calculating the Pearson correlation distance from the correlation matrix (Fig. 2a), as $\mathbf{P} = 1 - \mathbf{C}$, where $\mathbf{C}$ is the correlation matrix with all positive values.

Path-based encodings are in proximity of each other except for Daylight, which is linked to PubChem and FCFP, and RDKIT, which is only connected to Avalon. DFS is the fingerprint of this category that is most correlated within its category, reaching all other path-based algorithms in at most two steps within the MST. Circular and string-based fingerprints are mostly interconnected with each other, apart from LSTAR. MHFP connects with FCFP, ECFP and RAD2D, consistently with the fact that it also relies on circular fragments, while MAP4 connects with ASP, which likely reflects the fact that it encodes topological distances between atom pairs. FCFP is unique among all fingerprints, given that it connects with a fingerprint from all other categories except for pharmacophore-based encodings. This is especially surprising given that FCFP uses pharmacophoric information for the atom identifiers, which one might assume would lead to higher correlation with PH2 and PH3. Furthermore, it is notable that ECFP and FCFP correlate more strongly with MHFP than with each other, despite using the same algorithm except for the atom definitions. This seems to suggest that MinHashing SMILES substrings provide a hybrid representation that captures both chemical and pharmacophoric properties of the molecule. Substructure-based fingerprints are the most diverse, with only KR not connecting to algorithms belonging to different categories. PubChem and MACCS are linked to Daylight and AP respectively, while ESTATE is related to FCFP. This indicates that the fragment choices of these encodings are mostly orthogonal with each other and that, overall, this category is correlated to path-based and circular approaches. Pharmacophore fingerprints are separated from all other categories, consistently with the correlation matrix and their pairwise similarity distribution. The closest neighbor from a different class is AP,

which is connected to PH2, reflecting the fact that that both algorithms rely on distances between atom pairs.

Finally, this analysis confirms the assumption that, when deciding which fingerprint to use for similarity searches or QSAR modeling, the optimal strategy is to consider approaches belonging to different categories in order to minimize redundancy.

Similarity search ranking comparison

Similarity searching is often employed to identify the top K most similar compounds to a query molecule, e.g. to identify new bioactive molecules given a ligand for a protein of interest according to the similarity principle [59–61]. To examine whether different fingerprints would produce the same hits when used for similarity-based virtual screening, we repeated the sampling procedure described for the correlation comparison analysis and calculated for each compound the top 1% most similar molecules. We performed this procedure for each fingerprint and given a pair of encodings, we measured how many hits were ranked in the top 1% by both approaches. Finally, to evaluate whether natural products and drug-like compounds yield different results, we repeated this procedure for both the COCONUT and Drug Repurposing Hub datasets (Additional file 1: Figure S3).

Most fingerprint pairs exhibit an overlap score of approximately 25% on natural products, meaning that given a query molecule, 25% of the virtual screening hits are the same using both fingerprints. DFS and ASP show higher overlap than average (62%), consistently with the use of similar path enumeration algorithms to encode chemical graphs. When comparing the results obtained on COCONUT with the ones from Drug Repurposing Hub, the change in overlap percentage is between − 4% and 10% and is statistically significant for most fingerprint pairs (Additional file 1: Figure S3b). Finally, the ranking overlap difference is mostly consistent with the change observed in terms of similarity score correlation. For example, ESTATE and RAD2D fingerprints are generally more diverse from other encodings in the natural product space both in terms of top 1% ranking and overall pairwise Tanimoto correlation.

Exploring the natural product chemical space via dimensionality reduction

To analyze the effect that fingerprints have on capturing the distribution of NPs in the chemical space, we compared their bidimensional embeddings via UMAP (Fig. 3). Additionally, we investigated whether any embedding could separate NPs according to different taxonomical classes, given that different organisms produce biomolecules in different ranges of molecular weight, fraction of *sp3*-hybridized carbon and logP [20]. To do so, we colored the UMAP projections of NPs according to their taxonomy, after removing all compounds originating from multiple organisms.

Overall, no fingerprint can visually separate NPs according to their taxonomy, indicating that while different organism types generally produce compounds with different molecular properties, there is a significant overlap between these distributions. This is also consistent with the non-negligible fraction of NPs which are produced by multiple taxonomical classes found in COCONUT (4%).

Concerning the quality of the embeddings, PH2 and PH3 have atypical behaviors compared to all other fingerprints, with the former having one large compound group separated from everything else, while the latter showing none. These patterns are likely caused by the very broad similarity distribution observed for these fingerprints, making it difficult for the UMAP algorithm to preserve the manifold correctly.

Substructure-based fingerprints provide clear grouping of compounds according to their chemical structure, as shown by the clearly separated clusters in their embeddings, although this does not necessarily correlate with taxonomical information.

Path-based and circular fingerprints instead seem to provide much more uniform embeddings, causing most clusters to be closer together than for substructure-based approaches and making the manifold internal structure less distinct.

Finally, MAP4 and MHFP have comparable embeddings to path-based and circular fingerprints, albeit with a larger number of isolated compounds.

Classification performance

Depending on the classifier, metric and assay of interest, different fingerprints perform the best, with no clear favorite across the board. The only consistent pattern across all analyses is that pharmacophore fingerprints tend to underperform for classification, likely due to their inability to precisely distinguish chemical motifs.

When considering RF, in terms of global classification metrics, on average RAD2D achieves the best MCC (0.506), LSTAR outperforms all alternatives in terms of ROC-AUC (0.900) and MHFP performs the best in terms of PR-AUC (0.669), as shown in Additional file 1: Table S7. ASP is also a competitive option, ranking first in terms of ROC-AUC on 3 datasets out of 12 (Additional file 1: Table S9). In terms of individual datasets, LSTAR is especially promising for antiviral activity prediction (0.90 ROC-AUC, 0.71 PR-AUC), while MHFP excels at modeling the antitumor dataset (0.89 ROC-AUC, 0.82 PR-AUC). To further inspect the classification behavior of each fingerprint, we visualized their performance

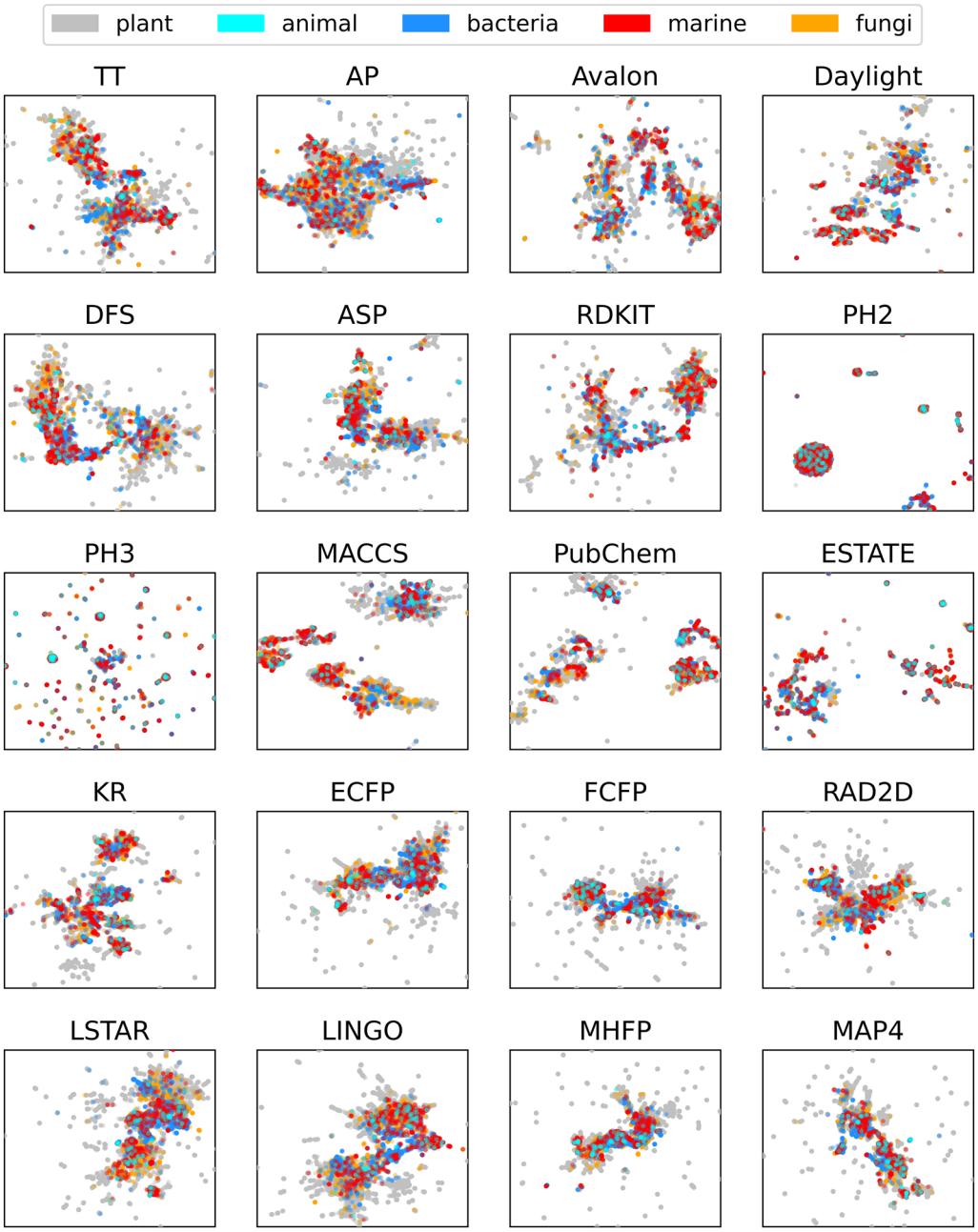


Fig. 3 Plot of UMAP embeddings for each fingerprint. Chemicals are colored on the basis of their source organism

in terms of precision, recall and specificity scatter plots (Fig. 4a and b), with contour lines indicating F1 score and balanced accuracy respectively. From these plots, we can conclude that MAP4, MHFP and LSTAR tend to have less false positives, while PubChem, MACCS and ESTATE generate less false negatives. Substructure fingerprints also rank particularly highly in terms of balanced accuracy (Fig. 4b), achieving a good balance of recall and specificity. When considering the post-hoc pairwise comparison tests, the situation differs from metric to metric (Additional file 1: Figure S5). Most fingerprints have statistically significant differences when considering precision, recall and specificity, while they are more comparable in terms of MCC, ROC-AUC and PR-AUC. This indicates that the false positive and true positive rate of RF models is significantly affected by the choice of molecular encoding, while the overall classification performance is less influenced.

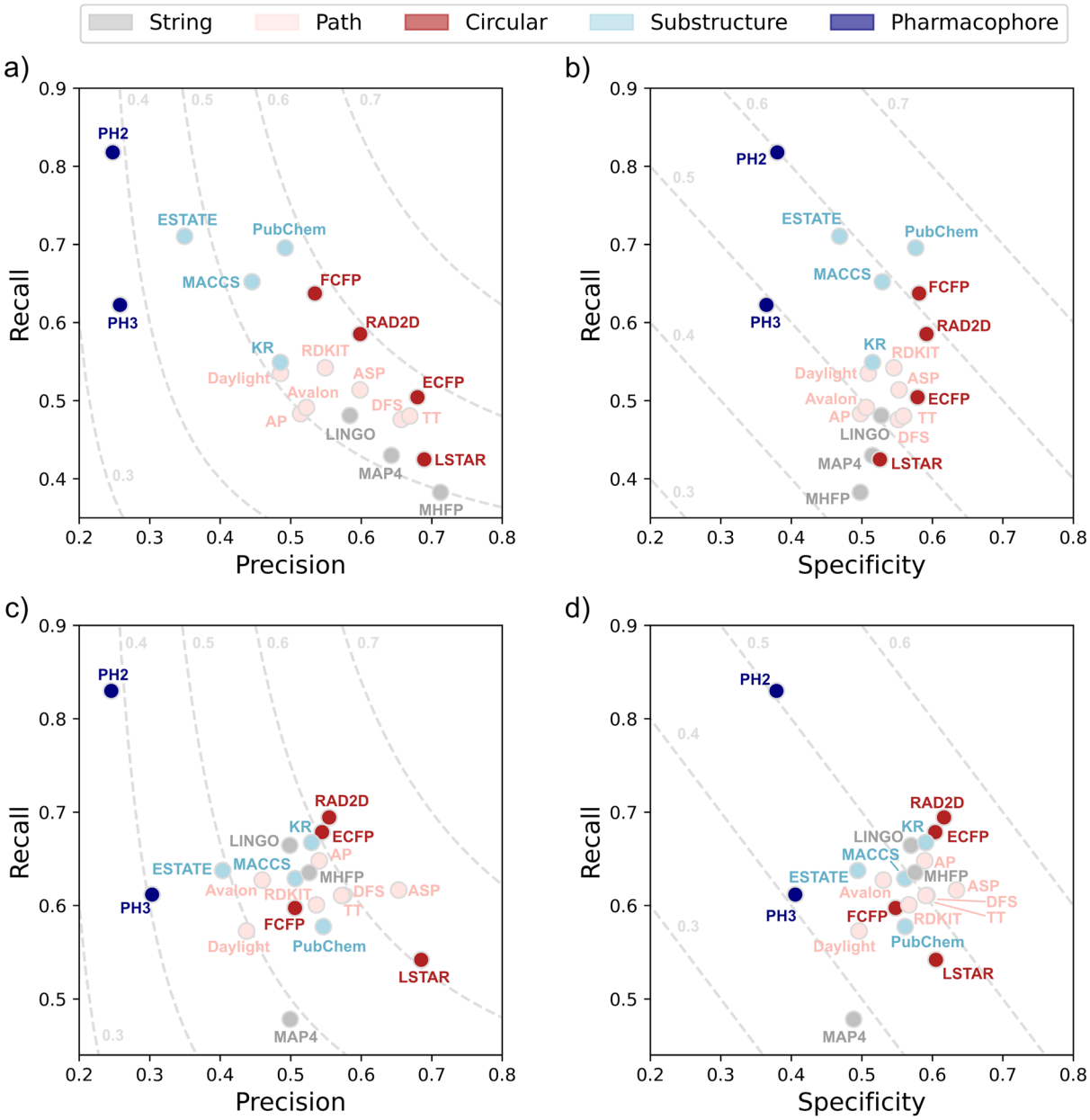


Fig. 4 Mean classification performance of each fingerprint across all datasets. **a** Recall versus precision plot for Random Forest, contour lines denote F1 scores. **b** Recall versus specificity plot for Random Forest, contour lines indicate balanced accuracy. **c** Recall versus precision plot for Dense Neural Networks, contour lines denote F1 scores. **d** Recall versus specificity plot for Dense Neural Networks, contour lines indicate balanced accuracy

When considering DNNs, ASP achieves the best MCC (0.562), ROC-AUC (0.8787) and PR-AUC (0.713), as shown in Additional file 1: Table S8. LSTAR is also a promising alternative, ranking first for anti-inflammatory activity modeling (0.96 ROC-AUC, 0.74 MCC) and achieving the highest precision in 3/12 datasets (Additional file 1: Table S10). One interesting difference between DNN and RF is the change in behavior of substructure-based fingerprints: while they generally lead to high recall for RF, they have more diverse performance when using DNNs. For example, PubChem here scores highly in precision, while ESTATE maintains high recall instead (Fig. 4c and d). One notable similarity between RF and DNN is that both have good performance with the MHFP fingerprint (Additional file 1: Figure S6). Given that its bit values are categorical, the expectation would be that this fingerprint would be a poor encoding choice for QSAR modeling with DNNs, since they generally

assume feature cardinality. In light of these results, it is likely that the performance could be increased even further with additional preprocessing, e.g. one-hot encoding of categorical bits. Finally, when considering the post-hoc statistical tests, all methods are equal in terms of recall, while there are many significant differences in PR-AUC compared to RF (Additional file 1: Figure S6).

Conclusions

Natural products are a promising class of compounds for drug discovery which is steadily becoming a crucial focus for biomedical research, thanks to their structural diversity, potency and selectivity in biological pathways. However, the best practices for molecular featurization of natural products is still an open question, given how different they are from typical drug-like molecules, thus limiting their use in cheminformatics applications.

Our analysis of molecular fingerprints in the natural product chemical space shows that algorithms belonging to the same category tend to be highly correlated, but they strongly diverge in terms of classification performance, pairwise similarities and chemical space representation when comparing them across categories. This finding suggests that when choosing which encoding to use for cheminformatics applications, it is beneficial to sample multiple fingerprints belonging to different classes to maximize diversity.

Concerning bioactivity prediction, our results show that the choice of molecular fingerprint has a significant impact on the classification performance across datasets (Additional file 1: Table S11). While ECFP has been the de-facto standard fingerprint for encoding drug-like compounds, our analysis indicates that other encodings can match or outperform them—the most promising ones being ASP, LSTAR and MHFP. Additionally, we highlight that while some approaches tend to perform better than others, no encoding significantly outperforms all others across all QSAR datasets in our study. This finding indicates that it is necessary to evaluate multiple fingerprints in order to obtain the best performance possible when constructing molecular property prediction models for the NP chemical space.

In terms of further fingerprint development, our study highlights two key findings. First, substructure-based fingerprints can be competitive with path and circular algorithms on NP modeling, even though they were developed for different types of molecules. As such, it would be interesting to specifically create substructure-based encodings for NPs, considering the most frequent motifs of NPs. The recently developed Natural Compound Molecular Fingerprints (NC-MFP) could be an interesting starting point for the investigation of substructure-based approaches for this class of compounds. [62]

Second, different graph traversal algorithms lead to substantially different fingerprints in terms of QSAR performance. As such, it would be interesting to pair new atom identifiers or fragment encoding algorithms with the most promising path and circular fingerprints. One particularly intriguing possibility would be to use data-driven approaches to process SMILES substrings obtained by e.g. LSTAR or ASP, potentially combining the robustness of expert-defined encodings with the expressiveness of learned molecular representations.

Scientific contribution statement

This work is to our knowledge the first benchmarking study of molecular fingerprints for similarity searches and bioactivity prediction on natural products, a biologically relevant class of compounds that has seen limited cheminformatics modeling so far. Crucially, our findings indicate that Extended Connectivity Fingerprints, the most common encoding for drug-like compounds, can be outperformed by other molecular fingerprints, highlighting the importance of evaluating multiple encoding approaches and suggesting new research directions. Finally, we provide an open-source Python package to compute all molecular fingerprints investigated in this study to streamline their use in further cheminformatics applications.

Abbreviations

QSAR	Quantitative Structure–Activity Relationship
ECFP	Extended Connectivity Fingerprint
MHFP	MinHash Fingerprint
NP	Natural product
TT	Topological Torsion fingerprint
AP	Atom Pair fingerprint
DFS	Depth First Search fingerprint
ASP	All Shortest Paths fingerprint
PH2	Pharmacological Pairs fingerprint
PH3	Pharmacological Triplets fingerprint
FCFP	Functional Class Fingerprint
KR	Klekotha-Roth fingerprint
MAP4	MinHashed Atom Pair fingerprint
MCC	Matthews Correlation Coefficient
ROC-AUC	Receiver Operating Characteristic Area Under Curve
PR-AUC	Precision Recall Area Under Curve
PCA	Principal Component Analysis
MST	Minimum Spanning Tree
UMAP	Unifor Manifold Approximation and Projection
RF	Random Forest
DNN	Dense Neural Network

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-024-00830-3.

Additional file 1: Table S1. Number of compounds that were retained after each preprocessing step. Chemical structure validity was assessed via RDKIT and the ChEMBL structure curation package.1,2 Taxonomy validity was evaluated by checking whether the source organism

information contained any predefined keywords, as done in a previous study by Capecchi et al. **Table S2.** Class distribution of each batch of the preprocessed subset of the COCONUT database used in this study. **Table S3.** Murcko scaffold diversity for each batch of the preprocessed subset of the COCONUT database used in this study. **Table S4.** P-values for the Mann Whitney tests with Benjamini-Hochberg correction between the similarity score distributions arising from the COCONUT and Drug Repurposing Hub datasets for each fingerprint. **Table S5.** Fingerprint saturation percentage for the COCONUT and Drug Repurposing Hub datasets. **Table S6.** Pearson correlation between using count or binary bits for a given fingerprint on the COCONUT and Drug Repurposing Hub datasets. P-values are calculated according to one-sample Mann Whitney tests with Benjamini-Hochberg correction. **Table S7.** Mean classification performance of each fingerprint using Random Forest across all datasets. **Table S8.** Mean classification performance of each fingerprint using a Dense Neural Network across all datasets.**Table S9.** Best performance rank counts for each fingerprint across all datasets for Random Forest. **Table S10.** Best performance rank counts for each fingerprint across all datasets for Dense Neural Networks. **Table S11.** Friedman test p-values evaluating the presence of significant differences in the performance of fingerprints across all datasets. **Figure S1.** Jaccard-Tanimoto similarity distribution for each fingerprint across all possible pairwise comparisons in the Drug Repurposing Hub dataset. Violin plots indicate the percentiles of the distribution of Jaccard-Tanimoto similarities, with the circle indicating the median similarity value.**Figure S2.** Correlation matrix of all pairwise similarities for all fingerprints evaluated in this study on the Drug Repurposing Hub dataset.**Figure S3.** Similarity search ranking overlap between fingerprints, focusing on the top 1% most similar compounds. **a** Rank overlap between fingerprints on the COCONUT dataset. **b** Difference in rank overlap between fingerprints when comparing the values obtained on the COCONUT and Drug Repurposing Hub datasets. Positive overlaps mean that a given fingerprint pair has a higher overlap on natural products than on drug-like compounds. Asterisks denote significance (α=0.05) according to a one-sample Mann Whitney U test with Benjamini Hochberg correction. Raw p-values are available on the Github repository of this article.**Figure S4.** Significance of the Random Forest performance differences between fingerprint pairs across all datasets, according to a 2-tailed Wilcoxon test with the Benjamini-Hochberg correction. Red denotes whether the difference is significant (α=0.05).**Figure S5.** Significance of the Dense Neural Network performance differences between fingerprint pairs across all datasets, according to a 2-tailed Wilcoxon test with the Benjamini-Hochberg correction. Red denotes whether the difference is significant (α=0.05).**Figure S6.** Performance comparison for each fingerprint depending on the classifier. The x-axis shows the mean ROC-AUC performance of a Random Forest classifier trained with a given fingerprint. The y-axis shows the mean ROC-AUC performance of a Dense Neural Network using different fingerprints as inputs.

Acknowledgements
D.Bo. thanks Dr. Isabel Wilkinson for the helpful discussion on the scope of the project, Maximilian Schuh for his feedback on the code, Daniela Koch, Ester Pachyn and Joshua Hesse for their input on the figures.

Author contributions
Conceptualization: D.Ba., V.C., R.T., D.Bo, F.G. Data curation: D.Bo. Formal analysis: D.Bo, D.Ba., V.C., R.T. Methodology: D.Bo., D.Ba, V.C., R.T., F.G. Software: D.Bo. Writing—original draft: D.Bo., D.Ba. Writing—review and editing: all authors. All authors have given approval to the final version of the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials
The Python package to compute all the fingerprints, as well as the classification metrics for each individual QSAR dataset and scripts necessary to reproduce the results presented in this study are available at https://github.com/dahvida/NP_Fingerprints.

Competing interests
TThe authors declare no competing financial interests.

Received: 20 December 2023 Accepted: 17 March 2024
Published online: 25 March 2024

References

1. Atanasov AG, Zotchev SB, Dirsch VM, Supuran CT (2021) Natural products in drug discovery: advances and opportunities. Nat Rev Drug Discov 20(3):200–216. https://doi.org/10.1038/s41573-020-00114-z
2. Chen Y, Kirchmair J (2020) Cheminformatics in natural product-based drug discovery. Mol Inform 39(12):2000171. https://doi.org/10.1002/minf.202000171
3. Mullowney MW, Duncan KR, Elsayed SS, Garg N, van der Hooft JJJ, Martin NI, Meijer D, Terlouw BR, Biermann F, Blin K, Durairaj J, Gorostiola González M, Helfrich EJN, Huber F, Leopold-Messer S, Rajan K, de Rond T, van Santen JA, Sorokina M, Balunas MJ, Beniddir MA, van Bergeijk DA, Carroll LM, Clark CM, Clevert D-A, Dejong CA, Du C, Ferrinho S, Grisoni F, Hofstetter A, Jespers W, Kalinina OV, Kautsar SA, Kim H, Leao TF, Masschelein J, Rees ER, Reher R, Reker D, Schwaller P, Segler M, Skinnider MA, Walker AS, Willighagen EL, Zdrazil B, Ziemert N, Goss RJM, Guyomard P, Volkamer A, Gerwick WH, Kim HU, Müller R, van Wezel GP, van Westen GJP, Hirsch AKH, Linington RG, Robinson SL, Medema MH (2023) Artificial intelligence for natural product drug discovery. Nat Rev Drug Discov. https://doi.org/10.1038/s41573-023-00774-7
4. Sorokina M, Merseburger P, Rajan K, Yirik MA, Steinbeck C (2021) COCONUT online: collection of open natural products database. J Cheminformatics 13(1):2. https://doi.org/10.1186/s13321-020-00478-9
5. Todeschini R, Consonni V (2009) Molecular descriptors for chemoinformatics methods and principles in medicinal chemistry, 1st edn. Wiley, Hoboken. https://doi.org/10.1002/9783527628766
6. Rodrigues T, Reker D, Schneider P, Schneider G (2016) Counting on natural products for drug design. Nat Chem 8(6):531–541. https://doi.org/10.1038/nchem.2479
7. Friedrich L, Cingolani G, Ko Y, Iaselli M, Miciaccia M, Perrone MG, Neukirch K, Bobinger V, Merk D, Hofstetter RK, Werz O, Koeberle A, Scilimati A, Schneider G (2021) Learning from nature: from a marine natural product to synthetic cyclooxygenase-1 inhibitors by automated de novo design. Adv Sci 8(16):2100832. https://doi.org/10.1002/advs.202100832
8. Siramshetty VB, Nguyen D-T, Martinez NJ, Southall NT, Simeonov A, Zakharov AV (2020) Critical analysis. J Chem Inf Model 60(12):6007–6019. https://doi.org/10.1021/acs.jcim.0c00884
9. Zhou Y, Cahya S, Combs SA, Nicolaou CA, Wang J, Desai PV, Shen J (2019) Exploring tunable hyperparameters for deep neural networks with industrial ADME data sets. J Chem Inf Model 59(3):1005–1016. https://doi.org/10.1021/acs.jcim.8b00671
10. Shen J, Nicolaou CA (2019) Molecular property prediction: recent trends in the era of artificial intelligence. Drug Discov Today Technol 32–33:29–36. https://doi.org/10.1016/j.ddtec.2020.05.001
11. van Tilborg D, Alenicheva A, Grisoni F (2022) Exposing the limitations of molecular machine learning with activity cliffs. J Chem Inf Model 62(23):5938–5951. https://doi.org/10.1021/acs.jcim.2c01073
12. O'Boyle NM, Sayle RA (2016) Comparing structural fingerprints using a literature-based similarity benchmark. J Cheminformatics 8(1):36. https://doi.org/10.1186/s13321-016-0148-0
13. Muegge I, Mukherjee P (2016) An overview of molecular fingerprint similarity search in virtual screening. Expert Opin Drug Discov 11(2):137–148. https://doi.org/10.1517/17460441.2016.1117070
14. *RDKit*. https://www.rdkit.org/. Accessed 9 May 2021.
15. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O, Torrance G, Evelo CT, Guha R, Steinbeck C (2017) The chemistry development kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. J Cheminformatics 9(1):33. https://doi.org/10.1186/s13321-017-0220-4
16. Hinselmann G, Rosenbaum L, Jahn A, Fechner N, Zell A (2011) jCompoundMapper: an open source java library and command-line tool for

chemical fingerprints. J Cheminformatics 3(1):3. https://doi.org/10.1186/1758-2946-3-3
17. Capecchi A, Probst D, Reymond J-L (2020) One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. J Cheminformatics 12(1):43. https://doi.org/10.1186/s13321-020-00445-4
18. Probst D, Reymond J-L (2018) A probabilistic molecular fingerprint for big data settings. J Cheminformatics 10(1):66. https://doi.org/10.1186/s13321-018-0321-8
19. Lyu C, Chen T, Qiang B, Liu N, Wang H, Zhang L, Liu Z (2021) CMNPD: a comprehensive marine natural products database towards facilitating drug discovery from the ocean. Nucleic Acids Res 49(D1):D509–D515. https://doi.org/10.1093/nar/gkaa763
20. Capecchi A, Reymond J-L (2021) Classifying natural products from plants, fungi or bacteria using the COCONUT database and machine learning. J Cheminformatics 13(1):82. https://doi.org/10.1186/s13321-021-00559-3
21. Bento AP, Hersey A, Félix E, Landrum G, Gaulton A, Atkinson F, Bellis LJ, De Veij M, Leach AR (2020) An open source chemical structure curation pipeline using RDKit. J Cheminformatics 12(1):51. https://doi.org/10.1186/s13321-020-00456-1
22. Bemis GW, Murcko MA (1996) The properties of known drugs. 1. Molecular frameworks. J Med Chem 39(15):2887–2893. https://doi.org/10.1021/jm9602928
23. Corsello SM, Bittker JA, Liu Z, Gould J, McCarren P, Hirschman JE, Johnston SE, Vrcic A, Wong B, Khan M, Asiedu J, Narayan R, Mader CC, Subramanian A, Golub TR (2017) The Drug Repurposing Hub: a next-generation drug library and information resource. Nat Med 23(4):405–408. https://doi.org/10.1038/nm.4306
24. Riniker S, Landrum GA (2013) Open-source platform to benchmark fingerprints for ligand-based virtual screening. J Cheminformatics 5(1):26. https://doi.org/10.1186/1758-2946-5-26
25. Heikamp K, Bajorath J (2011) Large-scale similarity search profiling of ChEMBL compound data sets. J Chem Inf Model 51(8):1831–1839. https://doi.org/10.1021/ci200199u
26. Rohrer SG, Baumann K (2009) Maximum unbiased validation (MUV) data sets for virtual screening based on pubchem bioactivity data. J Chem Inf Model 49(2):169–184. https://doi.org/10.1021/ci8002649
27. Sorokina M, Steinbeck C (2020) Review on natural products databases: where to find data in 2020. J Cheminformatics 12(1):20. https://doi.org/10.1186/s13321-020-00424-9
28. Nilakantan R, Bauman N, Dixon JS, Venkataraghavan R (1987) Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. J Chem Inf Comput Sci 27(2):82–85. https://doi.org/10.1021/ci00054a008
29. Carhart RE, Smith DH, Venkataraghavan R (1985) Atom pairs as molecular features in structure-activity studies: definition and applications. J Chem Inf Comput Sci 25(2):64–73. https://doi.org/10.1021/ci00046a002
30. Gedeck P, Rohde B, Bartels C (2006) QSAR—how good is it in practice? Comparison of descriptor sets on an unbiased cross section of corporate data sets. J Chem Inf Model 46(5):1924–1936. https://doi.org/10.1021/ci050413p
31. Jarvis RA, Patrick EA (1973) Clustering using a similarity measure based on shared near neighbors. IEEE Trans Comput C–22(11):1025–1034. https://doi.org/10.1109/T-C.1973.223640
32. Ralaivola L, Swamidass SJ, Saigo H, Baldi P (2005) Graph kernels for chemical informatics. Neural Netw Off J Int Neural Netw Soc 18(8):1093–1110. https://doi.org/10.1016/j.neunet.2005.07.009
33. Mahé P, Ralaivola L, Stoven V, Vert J-P (2006) The pharmacophore kernel for virtual screening with support vector machines. J Chem Inf Model 46(5):2003–2014. https://doi.org/10.1021/ci060138m
34. Durant JL, Leland BA, Henry DR, Nourse JG (2002) Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci 42(6):1273–1280. https://doi.org/10.1021/ci010132r
35. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE (2021) PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res 49(D1):D1388–D1395. https://doi.org/10.1093/nar/gkaa971
36. Hall LH, Kier LB (1995) Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. J Chem Inf Comput Sci 35(6):1039–1045. https://doi.org/10.1021/ci00028a014
37. Klekota J, Roth FP (2008) Chemical substructures that enrich for biological activity. Bioinformatics 24(21):2518–2525. https://doi.org/10.1093/bioinformatics/btn479
38. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50(5):742–754. https://doi.org/10.1021/ci100050t
39. Bender A, Mussa HY, Glen RC, Reiling S (2004) Similarity searching of chemical databases using atom environment descriptors (MOLPRINT 2D): evaluation of performance. J Chem Inf Comput Sci 44(5):1708–1718. https://doi.org/10.1021/ci0498719
40. Vidal D, Thormann M, Pons M (2005) LINGO, an efficient holographic text based method to calculate biophysical properties and intermolecular similarities. J Chem Inf Model 45(2):386–393. https://doi.org/10.1021/ci049679 7
41. Bero SA, Muda AK, Choo YH, Muda NA, Pratama SF (2017) Similarity measure for molecular structure: a brief review. J Phys Conf Ser 892:012015. https://doi.org/10.1088/1742-6596/892/1/012015
42. McInnes L, Healy J, Melville J. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv. 2020. https://doi.org/10.48550/arXiv.1802.03426.
43. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction—umap 0.5 documentation*. https://umap-learn.readthedocs.io/en/latest/. Accessed 16 Oct 2023.
44. *DeepChem*. https://deepchem.io/. Accessed 11 Dec 2021.
45. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324
46. Kadra A, Lindauer M, Hutter F, Grabocka J. Well-Tuned Simple Nets Excel on Tabular Datasets. arXiv. 2021. https://doi.org/10.48550/arXiv.2106.11189.
47. Ballabio D, Grisoni F, Todeschini R (2018) Multivariate comparison of classification performance measures. Chemom Intell Lab Syst 174:33–44. https://doi.org/10.1016/j.chemolab.2017.12.004
48. Feng, Y.; Zhou, M.; Tong, X. Imbalanced Classification: A Paradigm-Based Review. arXiv June 30, 2021. http://arxiv.org/abs/2002.04592. Accessed 10 Oct 2022.
49. Haibo HE, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284. https://doi.org/10.1109/TKDE.2008.239
50. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. https://doi.org/10.1080/01621459.1937.10503522
51. Rey D, Neuhäuser M (2011) Wilcoxon-Signed-Rank Test. In: Lovric M (ed) International encyclopedia of statistical science. Springer, Berlin, Heidelberg, pp 1658–1659. https://doi.org/10.1007/978-3-642-04898-2_616
52. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B Methodol 57(1):289–300
53. Seabold S, Perktold J. Statsmodels: Econometric and Statistical Modeling with Python; Austin, Texas, 2010; pp 92–96. https://doi.org/10.25080/Majora-92bf1922-011.
54. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD (2015) Hyperopt: a python library for model selection and hyperparameter optimization. Comput Sci Discov 8(1):014008. https://doi.org/10.1088/1749-4699/8/1/014008
55. *PyTorch*. PyTorch. https://pytorch.org/. Accessed 6 Dec 2023.
56. Pedregosa, F. Scikit-Learn: Machine Learning in Python. *Mach. Learn. PYTHON* 6.
57. Flower DR (1998) On the properties of bit string-based measures of chemical similarity. J Chem Inf Comput Sci 38(3):379–386. https://doi.org/10.1021/ci970437z
58. Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. Proc Am Math Soc 7(1):48–50. https://doi.org/10.1090/S0002-9939-1956-0078686-7
59. Bender A, Glen RC (2004) Molecular similarity: a key technique in molecular informatics. Org Biomol Chem 2(22):3204. https://doi.org/10.1039/b409813g
60. Horvath D, Jeandenans C (2003) Neighborhood behavior of in silico structural spaces with respect to in vitro activity spaces—a novel understanding of the molecular similarity principle in the context of multiple receptor binding profiles. J Chem Inf Comput Sci 43(2):680–690. https://doi.org/10.1021/ci025634z
61. Ripphausen P, Nisius B, Bajorath J (2011) State-of-the-art in ligand-based virtual screening. Drug Discov Today 16(9–10):372–376. https://doi.org/10.1016/j.drudis.2011.02.011

62. Seo M, Shin HK, Myung Y, Hwang S, No KT (2020) Development of natural compound molecular fingerprint (NC-MFP) with the dictionary of natural products (DNP) for natural product-based drug development. J Cheminformatics 12(1):6. https://doi.org/10.1186/s13321-020-0410-3

Publisher's Note

Acknowledgements

First and foremost, I would like to thank my doctoral supervisor Prof. Dr. Stephan A. Sieber for giving me the opportunity to pursue a PhD and welcoming me into his research group. His insight, knowledge, humanity and boundless enthusiasm for research have been inspirational since our first meeting and I will always look back fondly at the time I spent as his doctoral student. Furthermore, I am deeply appreciative of his unwavering confidence in my ability as a researcher and the intellectual freedom he gave me to pursue my ideas. I am also grateful for the compassion and understanding he showed while I dealt with complicated personal circumstances. For all these reasons and more, he is to me a role model for leadership and research.

Next, I would like to thank Dr. Isabel Wilkinson for being my mentor throughout the doctorate. Since we met, I have always been impressed by her intelligence and brilliance as a researcher. I wish one day to be as knowledgeable as her in all matters of pharmacological research, from experimental biochemistry to cheminformatics. I am grateful for the complete and unconditional support she gave me during these years and the many valuable lessons on navigating the academic environment, scientific practice, professionalism and resilience. If I am the researcher I am today, it is thanks to her outstanding example and guidance.

I would like to express my gratitude to my industry collaborators from Merck KGaA, Dr. Daniel Kuhn and Dr. Lukas Friedrich. I am especially appreciative for your confidence in me as a scientist and for your industry insights, which played a vital role in our many collaborations throughout the doctorate. Thank you for showing me this different perspective on the field and for the many teachings on the use of cheminformatics for drug discovery, strongly broadening my horizon as a researcher.

I am also forever grateful for my academic partners Prof. Dr. Davide Ballabio, Prof. Dr. Francesca Grisoni, Prof. Dr. Viviana Consonni and Prof. Dr. Roberto Todeschini. Thank you for beginning my journey in cheminformatics and machine learning, thanks to the many afternoon lessons during the Bachelor and for giving me the opportunity to carry out my research project at Procter&Gamble. I am also sincerely grateful for your friendship and for all the help you've given me since then, e.g. for always being available to discuss projects and for the great time and the conference in Padova. I will always strive to live up to the excellent example you have set of what a researcher and mentor should be.

While a doctorate is all about science, no good research can happen without an amazing group environment, and no doctoral student can make it through the end without the support of friends and loved ones. In both aspects, I am beyond blessed, and I have so many people to thank for that.

The Pandas queens, Isabel and Dominik. Thank you so much for all the amazing dinners, cakes, trips and moments we shared during these 3.5 years. Our holiday to Iseo was so much (needed) fun and I'm always going to remember the hike where we ate burrata on top of a mountain and almost died of dehydration and heat stroke. Dominik, thank you for all the fun we had during the OC Praktikum supervision, I could not have asked for a better partner for that. You are incredible friends and I will always strive to be as kind and considerate as you are.

The original horror night crew, Markus and Martin. Thank you for worsening my horror movie addiction, the parties, the basketball games and the nice evenings we had during my time in Munich. I have so many nice memories of the time we spent together and I am so grateful for all the times you cheered me up during the ups and downs of the doctorate. I am extremely appreciative for our friendship, and I hope that regardless of where we will end up in the future, we will hang out again in the future.

The Computer Wizards, Martin and Maxi. God knows how many outright unsalvageable and tragicomical situations we have encountered as the IT team during these 3 years and a half, but somehow we made it work, sort of, kind of. I definitely would not have been able to do this group job without you, and I would not have wanted to do it with anyone else. I am truly thankful for your frankly insane expertise with everything hardware and LRZ related, and for turning even the most obnoxious situations in something we can laugh about.

My fellow haters of a certain hardware company that shall not be named, Maxi and Joshua. You are outstanding machine learning researchers and I really learned a lot by working with you. I am proud and thankful for all the collaborations we have had, and I hope we will work together again in the future. Most importantly, you are also amazing friends. I am also extremely thankful for all the memes and good times we had during these years, e.g. Maxi's stolen shoes in Croatia and birds in unusual predicaments.

I would also like to thank all the colleagues at the Sieber group I had the pleasure of meeting throughout the doctorate. While saying "thank you all for the nice environment" has become a sort of ritualistic expression to be added at the end of a progress talk, my time in the group has truly been amazing and I was really blessed to work in such a nice environment. In no particular order, thank you Jan, Thomas G., Stuart, Martin P., Angela, Till, Isabel, Seppi, Micha, Michi,

Yassmine, Franzi, Didi, Max, Thomas S., Joshua, Maxi, Dominik, Tao, Wei, Konstantin, Markus, Alex, Laura, Katrin, Martin K., Nina, Marianne, Sara. Babsi, Robert, Eric and Sylvia.

A special thank you goes also to Riccardo, Leo and Carlo. We might live several hundreds of kilometers apart, but distance does not change in the slightest the depth and importance our friendship has for me, and how crucial it was to make it through the lowest points of the doctorate. I might be an only child, but I still have three brothers. Thank you for everything.

Mamma, Papà. Questa sezione la scrivo in Italiano perché ci tengo che possiate leggere esattamente le mie parole senza traduttore. Non posso fare altro che ringraziarvi di cuore per tutto. Tutto il supporto che mi avete dato durante questi anni, tutto l'affetto che ho ricevuto ogni volta che sono tornato a trovarvi, tutta la resilienza che ogni giorno dimostrate nell'affrontare qualsiasi problema. Siete la mia roccia e siete la mia forza.

Last but definitely not least, I simply would not be where I am today personally and academically without my amazing partner Daniela. You are my compass, and words cannot convey how thankful I am to you for leading me to the love and joy we share in our life together. My German is unfortunately not the best, but it is sufficient to say that "Du bist meine Lebensgefährtin".