Machine Learning Methodologies for Interpretable Compound Activity Predictions

Kumulative Dissertation

zur Erlangung des Doktorgrades (Dr. rer. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Raquel Rodríguez Pérez

aus Barcelona, Spanien

Bonn

November, 2019

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftliche Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Referent: Univ.-Prof. Dr. rer. nat. Jürgen Bajorath
2. Referent: Univ.-Prof. Dr. rer. nat. Holger Fröhlich
Tag der Promotion: 20.02.20
Erscheinungsjahr: 2020

Abstract

Machine learning (ML) models have gained attention for mining the pharmaceutical data that are currently generated at unprecedented rates and potentially accelerate the discovery of new drugs. The advent of deep learning (DL) has also raised expectations in pharmaceutical research. A central task in drug discovery is the initial search of compounds with desired biological activity. ML algorithms are able to find patterns in compound structures that are related to bioactivity, the so-called structure-activity relationships (SARs). ML-based predictions can complement biological testing to prioritize further experiments. Moreover, insights into model decisions are highly desired for further validation and identification of activity-relevant substructures. However, the interpretation of complex ML models remains essentially prohibitive. This thesis focuses on ML-based predictions of compound activity against multiple biological targets. Single-target and multi-target models are generated for relevant tasks including the prediction of profiling matrices from screening data and the discrimination between weak and strong inhibitors for more than a hundred kinases. Moreover, the relative performance of distinct modeling strategies is systematically analyzed under varying training conditions, and practical guidelines are reported. Since explainable model decisions are a clear requirement for the utility of ML bioactivity models in pharmaceutical research, methods for the interpretation and intuitive visualization of activity predictions from any ML or DL model are introduced. Taken together, this dissertation presents contributions that advance in the application and rationalization of ML models for biological activity and SAR predictions.

Contents

List of abbreviations

1D, 2D, 3D	One-, two-, three-dimensional
ADMET	Absorption, distribution, metabolism, excretion, and toxicity
CNN	Convolutional neural network
DL	Deep learning
DNN	Deep neural network
DT	Decision tree
ECFP	Extended-connectivity fingerprint
HTS	High-throughput screening
MACCS	Molecular access system
ML	Machine learning
MMP	Matched molecular pair
MT	Multitask
QSAR	Structure-activity relationship
QSPR	Structure-property relationship
RECAP	Retrosynthetic combinatorial analysis procedure
RF	Random forest
SAR	Structure-activity relationship
SMILES	Simplified molecular-input line-entry system
SPR	Structure-property relationship
ST	Single-task
SV	Support vector
SVM	Support vector machines
SVR	Support vector regression
Tc	Tanimoto coefficient
VS	Virtual screening

Motivation

Machine learning (ML) and, more recently, deep learning models have gained attention in pharmaceutical research due to the emergence of "big data" at different levels including medicinal chemistry.[1–3] The exploration of structure-activity relationships (SARs) represents a critically important task in medicinal chemistry and is essential for the development of novel bioactive compounds.[4,5] ML models are suitable for leveraging and mining the nearly exponential increasing amounts of compound activity data that are currently generated and published.[6,7] ML enables qualitative or quantitative SAR modeling and the subsequent prediction of compound bioactivity from structural representations.[8,9] The identification of active compounds by computational methods plays an important role in drug discovery and complements high-throughput screening.[10] Virtual screening protocols can be implemented using ML so that experimental testing is prioritized on the basis of model predictions. Some small molecules specifically interact with multiple targets, which might cause higher drug efficacy or undesired side effects. Therefore, multi-target predictions or activity profile predictions are currently a fundamental challenge of high interest.[11] With the rise of deep learning techniques, the potential benefit of deep neural networks (DNNs) for bioactivity modeling requires a systematic assessment.[12,13] Hence, prediction scenarios that mimic real-life screening or introduce challenging test systems are required.[14] Moreover, despite being decisive for ML model quality, the influence of the nature of training data on activity predictions is still an underinvestigated issue. Finally, interpretable ML models would provide insights into structural patterns driving changes in predicted compound activity and enable the extraction of SAR information. However, ML-based predictions are difficult to rationalize, especially for DNNs, which are often considered as "black boxes".[13,15] Thus, insights into complex model decisions

are essentially prohibitive which often hinders the practical use of ML models in pharmaceutical research.[16] In this thesis, ML models are systematically developed, analyzed, and rationalized for the prediction of compound bioactivity against multiple targets. More specifically, the main objectives of this thesis are: (i) the application and comparison of ML strategies for the prediction of compound activity profiles, (ii) the study of the influence of training set conditions on model performance, and (iii) the improvement of the interpretability of ML-based compound activity predictions.

Thesis outline

This dissertation consists of eight chapters structured as follows. *Chapter 1* presents an introduction to drug discovery and important applications of ML models in pharmaceutical research, with emphasis on compound activity predictions. *Chapter 2* to *Chapter 7* contain six original publications representing the main work of this thesis. *Chapter 2* reports the development and benchmark of ML approaches for the prediction of compound profiling matrices. *Chapter 3* presents the ML-based classification of weakly and highly potent inhibitors against a panel of kinases. In *Chapter 4*, guidelines for training set size and composition are derived for support vector machines models applied to activity predictions. *Chapter 5* investigates the relative performance of single-target ML and multi-target DNNs for the prediction of multiple assays from screening data. *Chapter 6* systematically studies the feature importance in support vector machines models for activity and potency prediction. In *Chapter 7*, a method for the interpretation of activity predictions from any ML algorithm is introduced and validated. Finally, *Chapter 8* summarizes and discusses the major findings of this thesis.

Chapter 1

Introduction

Drug discovery

The ultimate goal of pharmaceutical research is the identification of novel compounds with desired properties for the treatment of a given disease. Drug discovery research broadly includes (i) target identification, (ii) hit discovery, and (iii) lead optimization.[4,17] A biological target, generally a protein, is involved in a dysfunctional biological process and its modulation alleviates symptoms or modifies the disease state.[18] The identification of putative therapeutic targets implies the understanding of the connection between the molecular mechanisms and the disease. Identified targets are subsequently validated to confirm the relationship between target and disease. Currently, two major groups of drug targets are G protein-coupled receptors and kinases.[19,20] Next, a search for active compounds (or hits) that bind to the target and modify its function is pursued. Hit identification mainly relies on high-throughput screening (HTS) technologies, which include miniaturized and robotized assay platforms that test the activity of thousands or hundreds of thousands of compounds against a biological target in a short time.[21,22] Once hits are found, lead compounds or classes that serve as starting points are obtained and undergo multi-parametric optimization to improve other desired properties. At this stage, absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties of lead compounds are characterized and improved.[4] Compound optimization consists of many cycles of synthesis of close analogs to improve the properties based on small structural modifications. After these discovery re-

search stages, a promising drug candidate must satisfy pre-clinical and clinical development requirements. Only if clinical evaluation in human cohorts is successful, a drug could be approved by the pharmaceutical regulatory agencies. **Figure 1** schematizes the main phases of drug discovery and clinical development of a new drug, which take an average time of 12 years and are very costly ($\sim$ \$2.6 billion according to a recent estimation).[23] Moreover, pharmaceutical R&D investment has considerably increased during the last years without a positive impact on the number of discovered drugs,[18] which reflects the existence of some issues and bottlenecks.[24]

Figure 1: Drug discovery and development. An overview of the main stages in drug discovery and development is shown.

The role of computational approaches

A variety of computational approaches have been introduced to improve and accelerate drug discovery.[25] Bioinformatics and chemoinformatics focus on data processing and thus differ from computational biology, which mathematically models and simulates biological systems, and computational chemistry, which has its foundation in theoretical and quantum chemistry. Bioinformatics typically studies "omic"-data such as genomics or proteomics, whereas chemoinformatics focuses on small molecules and their role as ligands.[26,27] As such, this thesis covers chemoinformatic approaches for drug discovery. Chemoinformatics was firstly defined as *"the mixing of information resources to transform data into information, and information to knowledge, for the intended purpose of making better decisions faster in the area of drug lead identification and optimization"*.[28] Bio- and chemoinformatics disciplines, which contribute to pharmaceutical research with considerable overlap, have been continuously evolving both with novel algorithms and applications.[29,30]

Machine learning in pharmaceutical research

Machine learning (ML) approaches have been established as essential tools for bio- and chemoinformatics.[3] There is a need for new data mining methods able to cope with growing amounts of heterogeneous data sets, which offer many opportunities but are difficult to analyze and utilize. ML belongs to the spectrum of artificial intelligence methods, which are closely linked to the big data era and have currently become a hot topic in many areas including pharmaceutical research.[31,32] ML uses statistical pattern recognition algorithms that enable a system to learn from experience and subsequently make predictions about new data. Supervised ML can be used to predict discrete or continuous variables, whereas unsupervised methods are mainly used for exploratory analysis, visualization and clustering. Particularly, deep learning (DL) is a subdiscipline of ML that encompasses non-linear methods which can model complicated relationships between input and output data using low-level representations.[33] DL has surpassed standard ML methods in disciplines such as computer vision and natural language processing.[34–36] Consequently, DL has also experienced an increasing interest in pharmaceutical research.[12,37,38] Both ML and DL have encountered applications across all stages of drug discovery and development.[16]

Models have yielded accurate predictions for distinct bioinformatics tasks including target[39,40] or biomarker discovery.[41] Some studies have shown the potential of ML methods to distinguish between cancer and non-cancer targets on the basis of gene expression[42] or predict the suitability of targets for drug development from physicochemical, structural and geometric features of protein cavities.[43] ML has also been used to predict drug response across cell lines on the basis of gene expression data.[44] DL offers opportunities to deal with large amounts of single-cell RNA sequencing data, reduce dimensionality, and identify cell-specific biomarkers or characterize cell states and types.[45–47] Using pre-clinical data, ML models also enabled the identification of gene signatures for patient sub-groups that respond better to drug treatment.[48] In addition, pathology image processing has experienced a considerable improvement after the introduction of DL methodologies which prevent the need of manually identifying "handcrafted" task-specific features.[49] DL models have been used

for the classification and segmentation of microscopy images[50] as well as the identification of breast cancer regions in a large data set of pathology images.[51]

Chemoinformatics has also benefited from ML modeling. Recently, computer-aided synthesis planning has become a relevant application[52] and some studies have focused on the prediction of the major reaction products given a set of reactant molecules[53] or the conditions of organic synthesis reactions.[54] For the task of novel chemical structures generation or *de novo* design, a variety DL methods, such as variational autoencoder,[55] generative adversarial networks,[56,57] recurrent neural networks,[58,59] and deep reinforcement learning,[60] have been recently proposed. These approaches seem promising for generating compounds with desired properties, but the chemical diversity and validity of output samples are currently still debated.

Since the chemical structure of a compound determines its properties, medicinal chemistry studies structure-property relationships (SPRs), which can be modeled using ML.[9] Distinct ML methods[61,62] including deep neural networks (DNNs)[63,64] have been applied to quantitative SPRs (QSPRs) modeling. One of the most relevant compound properties is biological activity. Compound activity predictions generally help at the hit identification stage, whereas QSPRs for potency or ADMET properties are often considered in lead optimization.[13,65]

Compound activity predictions

The understanding and analysis of structure-activity relationships (SARs) is a central goal in medicinal chemistry and drug discovery. Since the big data era has arrived in medicinal chemistry, influenced by HTS and combinatorial chemistry,[66,67] ML has become a method of choice to mine chemical information and find molecules with desired bioactivity. Going beyond volume, compound activity data fulfills other big data-related terms such as heterogeneity, confidence, complexity, variability, and veracity.[1] In this context, ML-based activity predictions have found some relevant applications.

Virtual screening

A key application of bioactivity prediction is virtual screening (VS), which complements HTS through a prioritization of experimental testing.[68,69] Despite the numerous experiments performed by HTS, few bioactive compounds are found and hit rates are typically below 1-2%.[70] Thus, VS aims at selecting small numbers of potentially active compounds from in-house, commercial or virtual combinatorial libraries.[71]

Aside from ligand-based methods, which use compound activity data for predictions and are at the heart of the research presented in this thesis, methods relying on structural information about the targets present alternative relevant approaches.[72] Structure-based VS requires the three-dimensional (3D) structure of the target macromolecule, which can be obtained by X-ray crystallography or nuclear magnetic resonance spectroscopy.[73] Molecular docking is the most popular structure-based approach, which aims to find the preferred orientation or binding conformation of a ligand to a receptor through a computationally intensive optimization.[74,75] Docking uses a mechanism to explore the space of protein-ligand geometries and a scoring function to rank the possibilities.[76] Recently, ML methods have been proposed to score protein-ligand interactions,[77] including convolutional neural networks.[78]

On the other hand, ligand-based VS requires active compound data and often becomes the method of choice when 3D structures are not available.[10] Similarity searching is the classical approach for the detection of active compounds based on known ligand data,[79] but ML models have become widely used.[80] Different studies have shown the ability of ML to identify structurally distinct compounds with similar activity (task also known as scaffold hopping),[81] which is a pre-requisite for successful VS.[82,83] In retrospective studies, Doddareddy *et al.* trained linear discriminant analysis and support vector machines (SVM) models on the basis of compound fingerprints and detected blockers of potassium ion channels potentially leading to cardiotoxic effects.[84] VS approaches have also shown successful results in prospective applications[85–87] including a support vector regression model that detected new inhibitors of histone deacetylase 1[88] and a naïve Bayes model which identified inhibitors of phosphatidylinositol 3-kinase.[89]

With the integration of experimental and computational screening, ML models trained on the basis of biological screening data might select less but "smarter" experiments for the next round.[10] Active learning approaches have been developed for VS,[90,91] where the model iteratively selects compounds to test and is updated with the acquired data. The choice of the next experiments can rely on exploration (i.e. selection of useful data for model building) or exploitation (i.e. selection of compounds likely to be active) strategies.[92] Nevertheless, HTS and VS integration is challenging due to the inherent noise, experimental variance, large data volumes, diversity of chemical classes, possible presence of distinct binding modes, as well as strong statistical imbalance between hits and inactive compounds in HTS data.[73]

Multi-target activity

ML models can be trained to predict multi-target activities of compounds, also referred to as activity profiles. **Figure 2** illustrates compound activity profiles for exemplary compounds and schematizes the difference between single-target and multi-target modeling. An important limitation of the "one target, one drug" paradigm is the non-consideration of multi-target activity at early stages of the drug discovery.[93,94] The ability of small molecules to specifically engage in interactions with multiple receptors is the molecular basis of polypharmacology. Therapeutic polypharmacology achieves a stronger therapeutic effect by simultaneously targeting distinct points in a particular pathogenic process.[11,95] and is promising to combat complex diseases, such as cancer or neurological disorders, that might require a more elaborate pharmacological action.[96] However, multi-target activities might also be responsible for undesired side effects. The ultimate objective of chemogenomics is fully characterizing the interactions between all available chemical ligands and biological targets, which is practically unfeasible.[71] Hence, chemogenomics focuses on the exploration and navigation of limited ligand-target spaces, typically on the basis of protein families or related receptors. Since compound bioactivity profiles would enable a better prioritization of drug candidates, activity prediction against multiple targets is a highly relevant topic.[97] However, the optimization of multi-target SARs is complicated.[11,95] Some approaches have been proposed

such as SVM modeling with distinct kernel functions that account for protein sequence, structure, and hierarchy information for compound-target binding prediction.[98] Moreover, the performance of chemogenomics models that predict the interaction or non-interaction of protein-ligand pairs has been assessed[91] and compared to individual SAR models.[99] Recently, DL architectures have also been applied to multi-target activity predictions.[100]

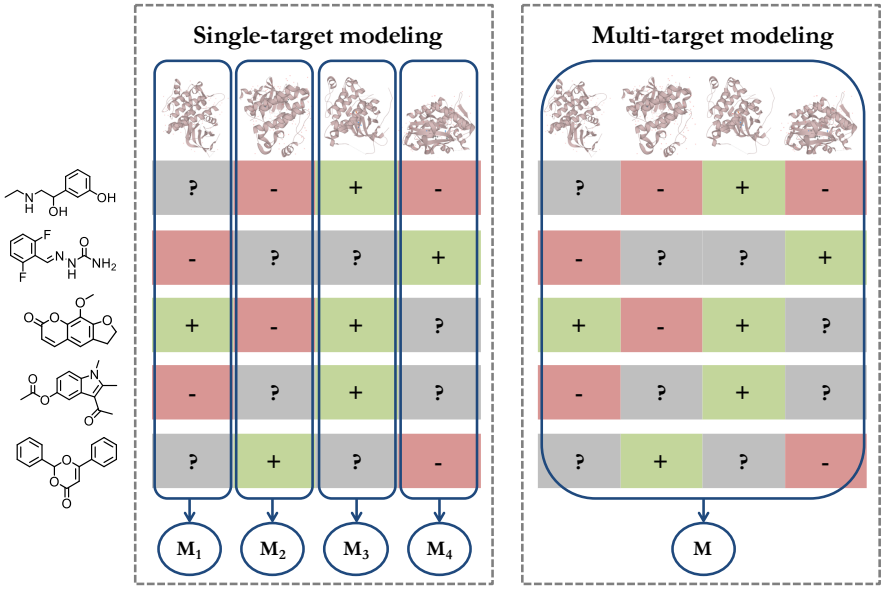


Figure 2: Single-target and multi-target modeling. A compound-target matrix with activity annotations (red: inactive, green: active, gray: unknown or missing) is schematized. Single-target (left) and multi-target (right) models predict compound bioactivity for one or multiple targets, respectively.

Orphan or novel targets

In principle, SAR modeling is not applicable to targets for which no actives are known, so-called orphan targets. However, some approaches have been implemented to overcome this limitation. The most simplistic method relies on ligands from homologous targets.[97] More sophisticated approaches have been proposed including SVM with specialized target-ligand kernels or linear combinations of SVM models.[101,102] Furthermore, chemogenomics models that discriminate between compound-target interacting or non-interacting pairs allow ligand binding prediction for orphan or novel targets.[103]

As stated above, ML and DL models are attractive for drug discovery because they enable handling large amounts of heterogeneous and noisy data. ML strategies have been successfully validated for many prediction tasks.[104] However, a "hype" is typically encountered when new technologies are first introduced to drug discovery[2] and this also applies to artificial intelligence methods. Hence, the real benefits of DL approaches remain unknown for many desired tasks. ML applications are often burdened by the lack of interpretability and rationalization of model success and failure, which is further aggravated in DL, given its extreme black box nature.[16,105] Even accurate DL models are often compromised in different application scenarios.[106] Therefore, if model decisions cannot be understood, the practical use of ML might be limited despite its undisputed potential.

Structure-activity relationships modeling

In addition to activity prediction, ML models statistically relate compound structure patterns to biological activity.[107] For SAR and quantitative SAR (QSAR) modeling, compounds are numerically represented by a feature vector and a learning algorithm maps these feature vectors to activity.[108] In particular, the chosen molecular representation defines the theoretical chemical space under study and the model accounts for similarity measures in such space.[109,110]

Molecular representations

Following graph theory concepts, chemical structures can be represented as graphs, in which atoms and bonds correspond to nodes and edges, respectively.[111] Linear notations such as the simplified molecular-input line-entry system (SMILES)[112] or the IUPAC international chemical identifier (InChI)[113] allow efficient storage of large compound data sets and can be converted to the molecular graph.[73] For predictive modeling, descriptors of molecular structure and properties are typically calculated either from the one-dimensional (1D) molecular formula, two-dimensional (2D) graph or 3D conformation.[109] Some examples of distinct complexity are molecular weight or atom counts (1D), connectivity indexes or structural fragments (2D), and van der Waals volume or

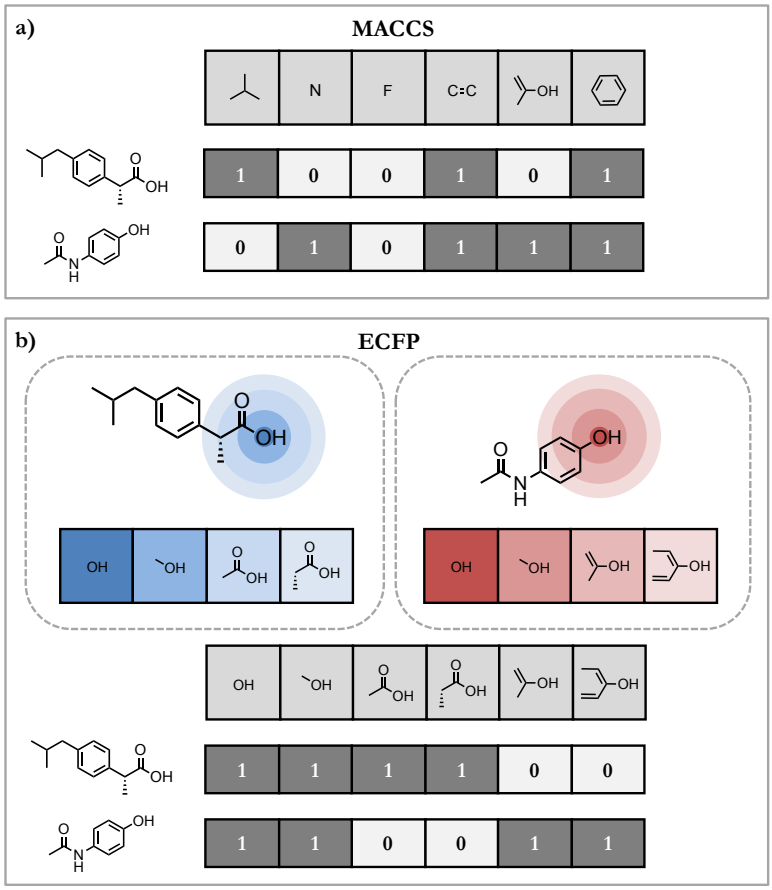


Figure 3: Molecular fingerprints. Schematic visualization of MACCS (a) and ECFP (b) for two exemplary compounds. These molecular fingerprints codify the presence (1) or absence (0) of chemical substructures or patterns. MACCS includes pre-defined structural keys or substructures, and ECFP encodes atom circular environments for each compound. In (b) atom environments are only generated for one exemplary atom.

spatial pharmacophores (3D).[10] Molecular 2D fingerprints are a well-known representation that encodes the presence or absence of chemical substructures or patterns in a binary vector. Structural keys codify pre-defined chemical patterns or substructures of a fixed length.[73] Molecular access system (MACCS) keys are a prominent example consisting of 166 bit positions,[114] which provide an easy-to-rationalize fingerprint as shown in **Figure 3a**. The extended-connectivity fingerprint (ECFP) is a hashed fingerprint that encodes circular atom environments up to a given diameter, as illustrated in **Figure 3b**.[115] Therefore, ECFPs are more general than structural keys and result in a higher-dimensional representation. ECFP length is variable by design, but it can be transformed to

a fixed-length vector through modulo mapping (folding). Recently, DL architectures that directly learn from the compound SMILES[116] or 2D graphs have been reported,[117,118] which alleviate the feature engineering process.

Structural similarity

SAR modeling is based on the "similarity property principle" which states that *"compounds with similar chemical structure share similar properties"*.[119] Structural similarity has been intensively studied in chemoinformatics for the comparison of compounds and their properties, mainly bioactivity.[120,121] Nevertheless, a clear and consistent similarity assessment by computational methods is complicated due to the subjective nature of the concept.[122] The matched molecular pair (MMP) formalism is a chemically intuitive way to determine analogs.[123] A pair of compounds forming an MMP only differs by a structural modification at a single site, as shown in **Figure 4**. Therefore, an MMP consists of a common core or key fragment, and a chemical transformation. The fragmentation required for MMP generation can be based on retrosynthetic combinatorial analysis procedure (RECAP) rules.[124] RECAP fragmentation is computationally efficient and accounts for synthetic accessibility. RECAP-MMPs can be organized in molecular networks, where nodes and edges represent compounds and MMP relationships, respectively. As a result, each disjoint cluster of the network is considered a unique analog series.[125,126]

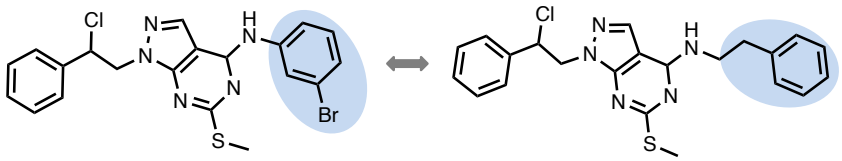


Figure 4: MMP concept. Exemplary MMP formed by two compounds with common core that only differ by a chemical transformation (highlighted in blue).

There are different metrics that can be used to quantify similarity (i.e. 1-distance) on the basis of molecular representations. Tanimoto coefficient (Tc) or Jaccard index is very popular for 2D fingerprints, and is given by (1) for two compound fingerprints.[79] Here A and B represent the sets of features present in either of the two molecules.

$$\mathrm{Tc}\,(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1}$$

Similarity searching

Similarity searching is the classic ligand-based approach for identifying new active compounds.[127,128] Following the similarity property principle, fingerprint similarity based on the Tc (or another metric) is used as an indicator of activity.[122] In particular, database molecules with unknown activity are ranked according to their decreasing similarity to an active reference molecule. Despite its simplistic nature, this approach is often effective providing an early enrichment of actives on the top of the ranking.[129] A variety of extensions of standard similarity searching have been introduced to improve performance including the combination of multiple searches (data fusion) or fingerprint modification.[73,130] Some exemplary methods are Turbo similarity searching,[131,132] consensus bit scaling,[133,134] and conditional correlated Bernoulli model.[135] With the advent of ML models, traditional similarity searching has mainly found its application in descriptive statistics, exploratory analysis and fast extensive calculations, such as large-scale VS.

Machine learning models for SAR analysis and prediction

The influence of chemical modifications on compound activity can be modeled using ML either in a qualitative (classification) or quantitative (regression) fashion. Based on their structural fingerprints, compounds are projected onto a well-defined m-dimensional chemical space. In such a feature space, similar molecules map closely together and ML models aim at recognizing differential patterns that enable accurate predictions. Formally, a supervised ML model relates a feature vector $\mathbf{x} = (x_1, ..., x_m) \in \mathcal{X}$ to an output label y, where $y = \{+1, -1\}$ for binary classification and $y \in \mathbb{R}$ for regression, through a function f so that $f(\mathbf{x}) = y$. The ML model attempts to minimize the expected test error, which can be decomposed into bias, variance and a constant irreducible

noise term, as shown in (2) for a test instance $\mathbf{x}$ with label y. Bias refers to the error introduced by approximating a real-life problem using a much simpler model. Variance measures the variability or sensitivity of the prediction function to a particular choice of data. Clearly, a successful model simultaneously achieves low bias and variance.

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

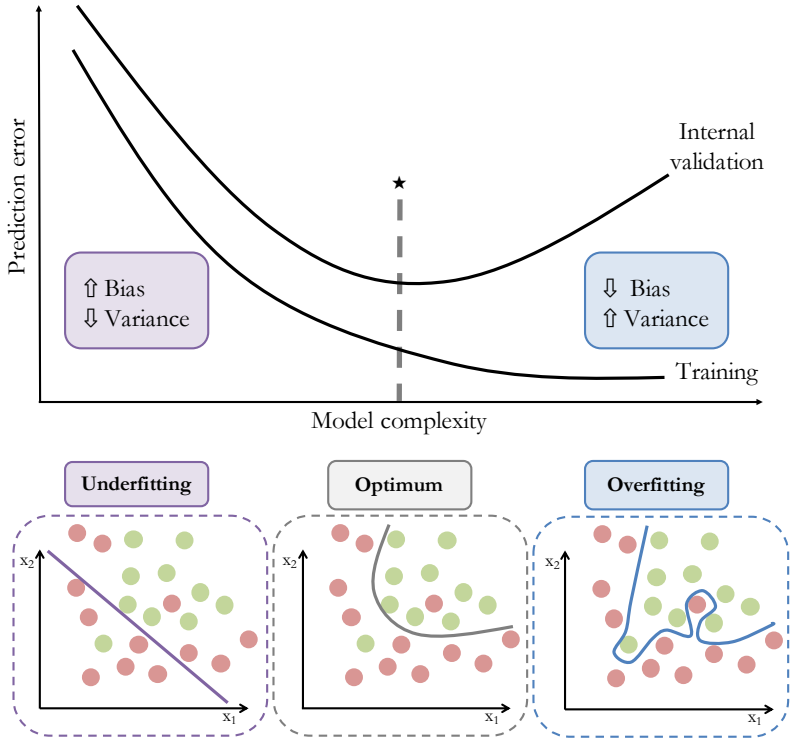


Figure 5: Optimization of model complexity. The prediction error in training and internal validation sets depends on model complexity. Very simple models, which generally have high bias and low variance, are not able to model the data properly (underfitting). Complex models are often characterized by low bias and high variance and model peculiarities of training data (overfitting). The optimum model complexity is obtained by minimizing the internal validation error.

As illustrated in **Figure 5**, model complexity determines the bias and variance trade-off, which reflects the need of complexity optimization when models rely on tuning hyper-parameters.[136] Simple models might be unable to capture the underlying patterns (underfitting) and complex models tend to fit the inherent noise of training data (overfitting). Validation is an essential part of model

building aiming at complexity optimization as well as performance estimation. Thus, it requires three data partitions: training set (model fitting), internal validation set (model selection), and test or external validation set (model assessment).[136,137] Cross-validation might be utilized to account for different splits or folds and make better use of the available data.[138]

Naïve Bayes

Naïve Bayes is a binary classifier that relies on the Bayes' theorem.[139] The term "naïve" refers to the assumption of conditional feature independence. Despite this simplification, naïve Bayes classifiers have been successfully applied in problems with correlated features.[140] Bayes' theorem is used to determine the probability that a compound $\mathbf{x}$ belongs to class y, i.e. $P(y|\mathbf{x})$. The likelihood function $P(\mathbf{x}|y)$ plays a central role. It can be estimated from the training set and related to the posterior probability through the Bayes' theorem (3).

$$P\left(y|\mathbf{x}\right) = \frac{P\left(\mathbf{x}|y\right) P\left(y\right)}{P\left(\mathbf{x}\right)} \tag{3}$$

$P(y)$ is the prior, i.e. either the known class probability distribution or its estimation over the training set, and $P(\mathbf{x})$ refers to the evidence, which acts as a normalization constant. Class likelihoods require an event model for feature distributions and, for binary fingerprints, features follow a multivariate Bernoulli distribution. For test predictions, the class with maximum likelihood estimate for the observed data $\mathbf{x}$ will be selected, as shown in (4).

$$y = \underset{\hat{y} \in \mathcal{Y}}{\operatorname{argmax}} P\left(\hat{y}|\mathbf{x}\right) \tag{4}$$

Random forest

A random forest (RF) consists of an ensemble of decorrelated decision trees (DTs) that elicit variance reduction.[141] A DT is a non-parametric method that infers a sequence of binary decision rules to split training data into subsets with better class separation. Splitting starts from the root (or top) node into child nodes using rules built on the basis of compound features. DT is a recursive

partitioning method where each child node might in turn split until a stopping criterion is reached. As illustrated in **Figure 6**, non-leaf nodes represent the decision rules, edges are possible outcomes, and the predominant class on the leaf nodes determines the prediction. A tree path from the root to the terminal node directly indicates the chain of feature-based decisions. DTs can capture complex interaction structures in the data and have a low bias, but it comes at the expense of high variance.[136]

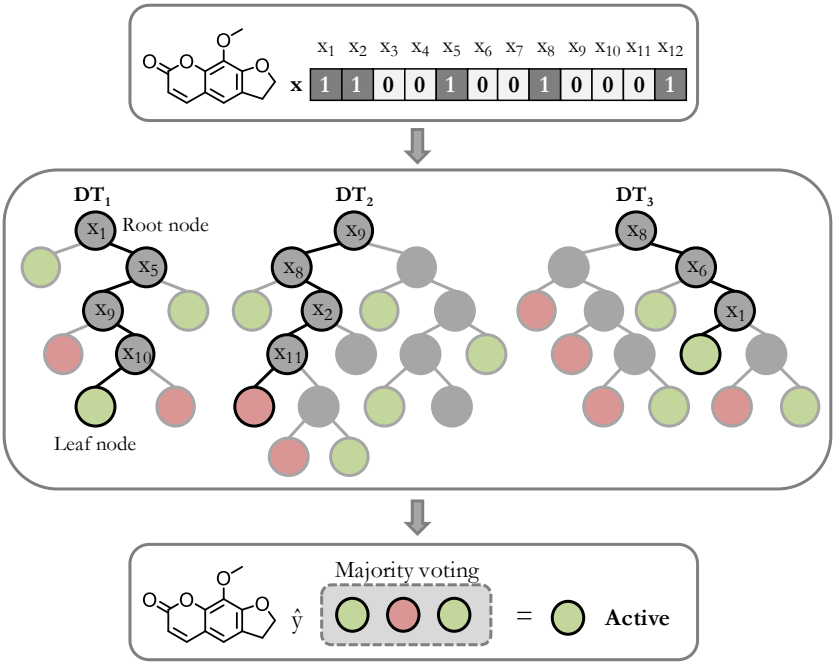


Figure 6: Principles of RF modeling. A compound with fingerprint **x** is predicted by a RF model with three DTs. Each non-leaf node represents a decision rule based on a feature (x_m). The tree path from the root to the leaf node is highlighted in black. The color of leaf nodes indicates the predicted label, e.g. active (green) or inactive (red). The final prediction for **x** is given by a consensus across individual tree predictions.

The classification and regression trees or CART algorithm[142] constructs DTs using feature thresholds yielding minimal Gini impurity values. This criterion encourages the formation of regions with high proportions of data assigned to one class,[139] and becomes zero when all instances in a node belong to the same class. G in a node τ is formally defined by (5), where C refers to the total number of classes and p_c to the probability of selecting a sample from class c.

$$G(\tau) = \sum_{c=1}^{C} p_c \left(1 - p_c\right) \tag{5}$$

RF builds a collection of individual DTs using random samples with replacement from the training set, which is known as bagging or bootstrap aggregating. Furthermore, RF randomly selects a feature set for node splitting (feature bagging) to prevent DTs relying on the same strong predictors.[141] These two bagging strategies decorrelate the trees and introduce variability within the ensemble model.[143] Final predictions are driven by a consensus across trees. RF model achieves reduced variance without increasing the bias of individual models.

Support vector machines

Classification

The SVM algorithm was initially proposed for binary classification. The SVM classifier attempts to find a hyperplane $H = \{\mathbf{x} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$, defined by a normal vector $\mathbf{w}$ and a bias b that separates the positive and negative classes.[144,145] For that purpose, SVM maximizes the margin, i.e. the distance between the closest training instances from each class and the hyperplane. These training instances are called support vectors (SVs). The hard-margin (or maximum margin) hyperplane can be obtained by minimizing the distance from H to the SVs of each class. However, said minimization problem has no solution when data is not separable by a linear function. Slack variables (ξ_i) are added to derive a soft-margin hyperplane that enables training errors.[146] To allow limited numbers of training compounds to map inside the margin or on the incorrect side of the hyperplane, the minimization problem can be formulated as indicated in (6).

$$\begin{gathered} \underset{\mathbf{w},b}{\text{minimize:}}\ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{subject to: } y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1 - \xi_i \text{ with } \xi_i \geq 0\ \forall i \end{gathered} \tag{6}$$

The cost or regularization hyperparameter C balances the magnitude of permitted training errors and the margin maximization. Small C values tolerate

larger errors, whereas large cost factors lead to a complex model. The primal optimization problem can be expressed in a dual form using Lagrange multipliers α_i, and the solution yields the normal vector of the hyperplane in (7). Training examples with non-zero α_i coefficients represent the SVs and solely determine the hyperplane position. These data points lie on the edge, within the margin or even on the incorrect side of the hyperplane. The binary SVM classification is schematized in **Figure 7**.

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{7}$$

Once the hyperplane is derived, test data are projected into the feature space. A ranking can be obtained using the real value $g(\mathbf{x}) = \sum_i \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b$, which geometrically corresponds to sliding the hyperplane from the most distant data point on the positive half space toward the negative side.[147] Test instances can be classified according to the side of the plane on which they fall, i.e. $f(\mathbf{x}) = \text{sgn}(g(\mathbf{x}))$, which means that compounds with $f(\mathbf{x}) = +1$ will be assigned to the positive class (e.g. active) and $f(\mathbf{x}) = -1$ to the negative class (e.g. inactive).

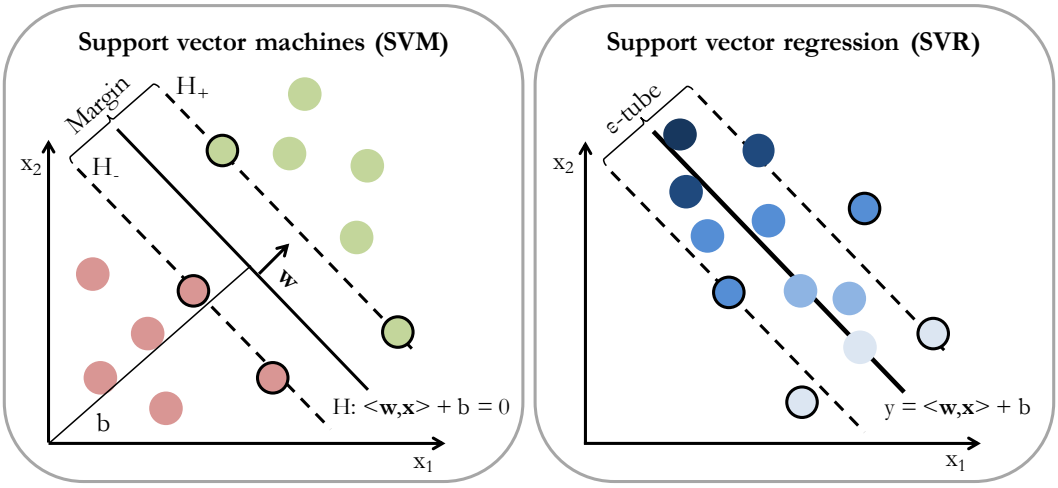


Figure 7: Principles of SVM and SVR modeling. In SVM (left), a hyper-plane is generated for binary classification by margin maximization. The goal is differentiating between active (green) and inactive (red) compounds. In SVR (right), a regression function is derived for real value prediction. The gradient from light to dark blue indicates increasing numerical values (e.g. potency). SVs are represented by a black circle and lie within the margin (SVM) or outside the ϵ-tube (SVR).

Regression

Support vector regression (SVR) is an extension of the SVM algorithm for real value predictions.[148] In this case, SVR aims at mapping $\mathbf{x}$ as close as possible to their real label y by deriving a function of the form $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.[149,150] Tolerated differences from observed and predicted values of training data are at most ϵ. However, slack variables are introduced to allow larger positive and negative deviations (ξ_i, ξ_i^*) from the so-called ϵ-tube.[149] In SVR, the hyperparameter C also controls the relaxation of error minimization problem and thus penalizes large slack variables. Analogously to classification, the optimization problem is formulated in (8) and can be solved using Lagrangian reformulation, giving the final regression function in (9).

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize:}}\ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i (\xi_i + \xi_i^*) \\ \text{subject to:}\ & y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon + \xi_i\ \forall i \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon + \xi_i^*\ \forall i \end{aligned} \quad \text{with } \xi_i, \xi_i^* \geq 0 \tag{8}$$

$$f(\mathbf{x}) = \sum_i (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i, \mathbf{x} \rangle + b \tag{9}$$

In SVR, SVs are the training data points that have either positive α_i or α_i^*. SVs lie on the ϵ-tube or outside of it and are the only training data used for the prediction of new test examples. **Figure 7** illustrates the generation of a SVR model.

Kernel trick

For nonlinear data that cannot be accurately modeled in the feature space $\mathcal{X}$, the scalar product $\langle \cdot, \cdot \rangle$ can be replaced by a kernel function $K(\cdot, \cdot)$.[146,151] Conceptually, the kernel function transfers the scalar product to a higher dimensional space $\mathcal{W}$ in which the data might be linearly separated, as shown in **Figure 8**. The advantage is that the non-linear mapping $\phi : \mathcal{X} \to \mathcal{W}$ does not need to be explicitly computed. The generation of the hyperplane or regression function only depends on SVs and not on the dimension of the input space, which allows calculations in a higher dimensional space. A variety of kernel functions exist including, among others, the Gaussian or radial basis function

kernel and the polynomial kernel. In addition, the Tanimoto kernel expression is shown in (10) for two compounds with fingerprints $\mathbf{x}_a$ and $\mathbf{x}_b$. This kernel function is based on the Tc and widely used in chemoinformatics.[152]

$$K_{\mathrm{Tc}}\left(\mathbf{x}_a, \mathbf{x}_b\right) = \frac{\langle \mathbf{x}_a, \mathbf{x}_b \rangle}{\langle \mathbf{x}_a, \mathbf{x}_a \rangle + \langle \mathbf{x}_b, \mathbf{x}_b \rangle - \langle \mathbf{x}_a, \mathbf{x}_b \rangle} \tag{10}$$

The kernel trick allows non-linear SVM and SVR models but confers "black box" character to the models.

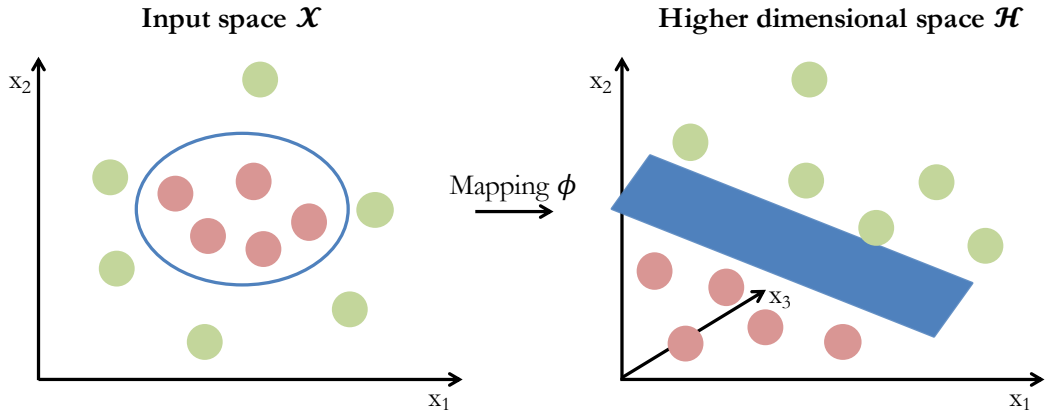


Figure 8: Kernel trick SVM applies a non-linear mapping (ϕ) to project molecular representations into a higher-dimensional ($\mathcal{W}$) space and enable a linear separation, when it is not possible in the original feature or input space ($\mathcal{X}$).

Deep learning

Deep neural networks

Feedforward deep neural networks (DNNs) are a series of functional transformations given by a collection of connected units that are organized in sequential layers.[153,154] Units in a layer act in parallel. These units are also referred to as neurons or basis functions. DNNs need to be constituted by at least an input layer, two hidden layers, and an output layer.[153] Each neuron receives inputs from units in the previous layer and computes its activation value, representing a vector-to-scalar function. The DNN schematic in **Figure 9** illustrates how each input to a node is modified by a unique set of weights and biases, thus giving unique combinations per activation. In particular, the neuron applies a

nonlinear activation function to the weighted sum of its inputs to generate its output.[139] The activation a_j^l of neuron j in layer l is given by (11), where σ is the activation function; w_{jk}^l indicates the weights at the hidden unit j of layer l; $a_k^{(l-1)}$ are the activations from the previous layer; b_j^l is the bias of the neuron; and the sum is over all neurons k in the layer $(l-1)$.[153]

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) = \sigma\left(z_j^l\right) \tag{11}$$

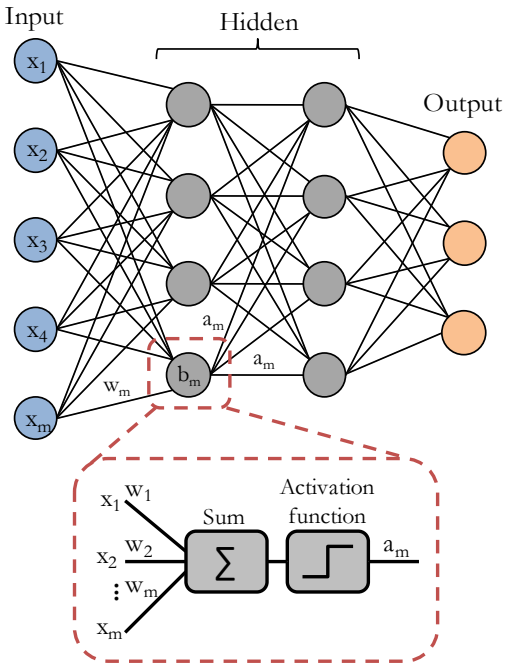


Figure 9: Principles of a DNN. The schematic representation of a DNN shows an input layer with five neurons (blue), two hidden layers with four neurons each (gray) and an output layer with three neurons (orange). Each edge has a unique weight w_i and each node has a unique bias b_i. Each neuron considers a weighted sum of the inputs x_i and applies an activation function to obtain the activation a_i, which is an input for the units of the next layer.

During the training phase, network weights and biases are modified so that the predicted output matches or approximates the correct label y. The cost function refers to the discrepancy between the output of the network and the real label and has to be minimized during training. Minor changes in weights and biases need to be related to small changes in the cost function, which is facilitated by the gradient of the cost function $\nabla C \equiv \left(\frac{\partial C}{\partial w_1}, \frac{\partial C}{\partial b_1}, ..., \frac{\partial C}{\partial w_l}, \frac{\partial C}{\partial b_l}\right)^T$.

The gradient defines the rate at which the cost will vary with respect to a change in the weights or biases.[153] Gradient descent methods can be used to minimize the cost $C(w, b)$, updating the weights and biases according to (12), where η is the so-called learning rate. This process is illustrated in **Figure 10**. The learning rate hyper-parameter needs to be small enough so that the approximation is accomplished but must also not be too small to avoid an extremely slow gradient descent process resulting in unfeasibly long training times.

$$\begin{aligned} w \to w' &= w - \eta \frac{\partial C}{\partial w} \\ b \to b' &= b - \eta \frac{\partial C}{\partial b} \end{aligned} \tag{12}$$

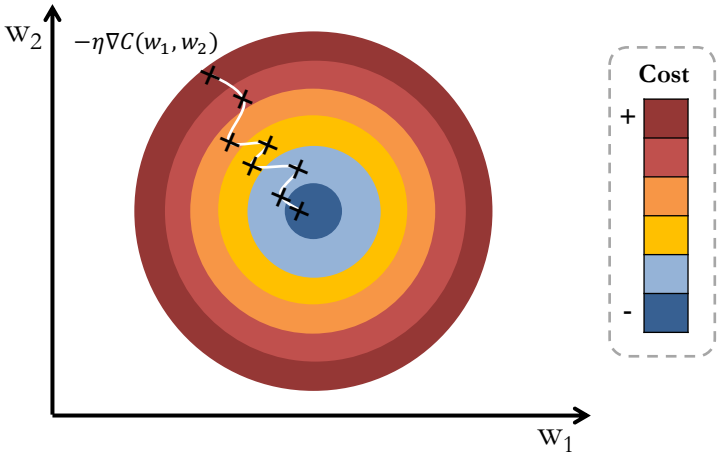


Figure 10: Gradient descent. The error surface is shown with respect to two weight components (w_1 and w_2), where the color gradient from blue to red represents increasing error or cost value. Since the gradient of the cost function indicates the direction of steepest ascent, the negative of the gradient is taken.

Backpropagation is an efficient method to compute the partial derivatives of the cost function with respect to any weight and bias in the network.[155] In practice, backpropagation applied to random training subsets or "mini batches" and the average is taken, which is known as stochastic gradient descent.[156] Backpropagation in combination with stochastic gradient descent gives an approximation of the cost function gradient that depends on all the training data, and weights and biases are updated accordingly. Cost and activation functions

that capture small changes in the weights and biases are required. For instance, cross-entropy is generally used as loss function in DNNs to calculate the distance between the predicted probabilities and the real labels, and rectified linear units are very popular in current DNN design. In addition, L2 and dropout regularization are the most common approaches to implicitly reduce the number of free parameters and thus prevent model overfitting.[153] Finally, the output layer determines whether a DNN is for binary, multi-class, multi-label classification or regression.[139]

Graph convolutional networks

Graph convolutional neural nets (CNNs) are another type of DNNs that have been extensively used in computer vision. CNNs search for a given pattern in different sections of the input matrix such as pixels in an image.[34] A filter or feature detector, which contains units with the same weight and bias parameters, is applied multiple times. Since CNNs enable representation learning as well as the mapping to the output, they have been recently applied to directly learn from the molecular graph.[117] Graph convolutional networks rely on the 2D compound graph to automatically generate molecular representations inspired by ECFP or the Morgan algorithm.[118] A CNN applies convolutions centered on atoms where the weights and biases are the learnable parameters to construct molecular representations. The convolution proceeds at different levels by considering contributions of neighboring atoms, which corresponds to the circular fingerprints concept of extending the radius of atom environments. Initially, a set of atom features summarizing the local atom environment (e.g. atom type, valence, formal charge or hybridization) and a neighbor list representing molecular connectivity are obtained for every atom.[157] Weight matrices and bias vectors are used to update the atom features, and pooling is applied using the maximum value. This process is typically repeated and finally a graph gathering layer is introduced to sum up all feature vectors across atoms. This final compound representation serves as input to a fully-connected layer. Hence, a CNN combines feature extraction and model building in a trainable model. This method is schematized in **Figure 11**. Representation learning enables the consideration of task-specific features and eliminates the necessity to pre-compute fixed compound descriptors.

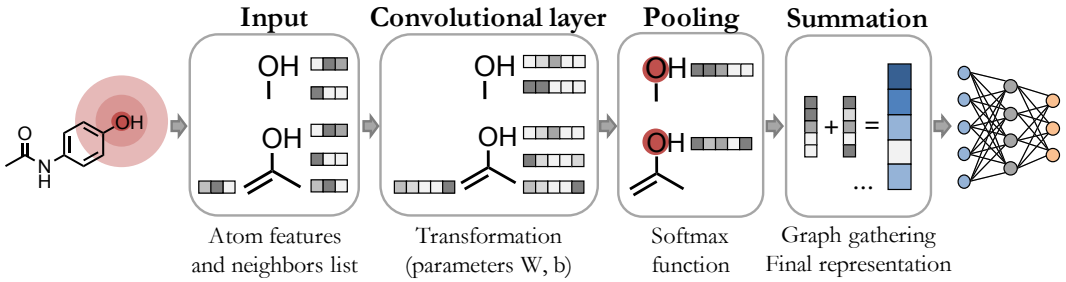


Figure 11: Fundamentals of a graph CNN. The process of representation learning is illustrated for a single central atom (O) and considering two neighbor levels, but in practice it is applied to all the graph nodes. Atom features (gray/white squares) and neighbors list (graph connectivity) are the inputs for the convolutional layer. Convolutions transform the inputs with a weight matrix W and bias b (learnable parameters) to obtain new feature vectors. In the pooling layer, a single vector is obtained per node (in the example, O atom) considering the maximum value for each feature across the neighbors. Finally, all feature vectors in the graph are summed up to obtain the final representation, which is the input for a fully-connected layer.

Modeling strategies

The prediction of compound bioactivity profiles represents a multi-label classification task, in which small molecules might bind to distinct targets. Models can learn on the basis of ligand data for a single target or multiple targets. Therefore, for the prediction of activity profiles, distinct modeling strategies can be considered. Single-task (ST) modeling builds one model per target and is often the default choice for multi-label classification. Accordingly, the multi-target problem is decomposed into distinct binary tasks and standard binary classifiers are applied, as shown in **Figure 12a**. Other strategies exist to transform the prediction problem into binary tasks, e.g. one-vs-one. On the other hand, some ML methods can be algorithmically modified or adapted to enable multitask (MT) learning.[158] MT learning simultaneously models the activity against multiple targets, as illustrated in **Figure 12b**. Many ML algorithms support multi-label classification, but some methods do not enable MT modeling with missing labels. For MT learning with DNNs, the output layer requires as many units as tasks (or targets). Hence, all tasks share network parameters and feature selection until they are submitted into separate classifiers at the output layer. For handling missing compound-target annotations, MT-DNN can be algorithmically adapted. Finally, *in silico* chemogenomic approaches or

proteochemometric models can be directly applied to interaction prediction.[159] Following this approach, ligand and protein descriptors are combined[160,161] and used as input for a binary model, which discriminates between interacting and non-interacting compound-target pairs, as schematized in **Figure 12c**.

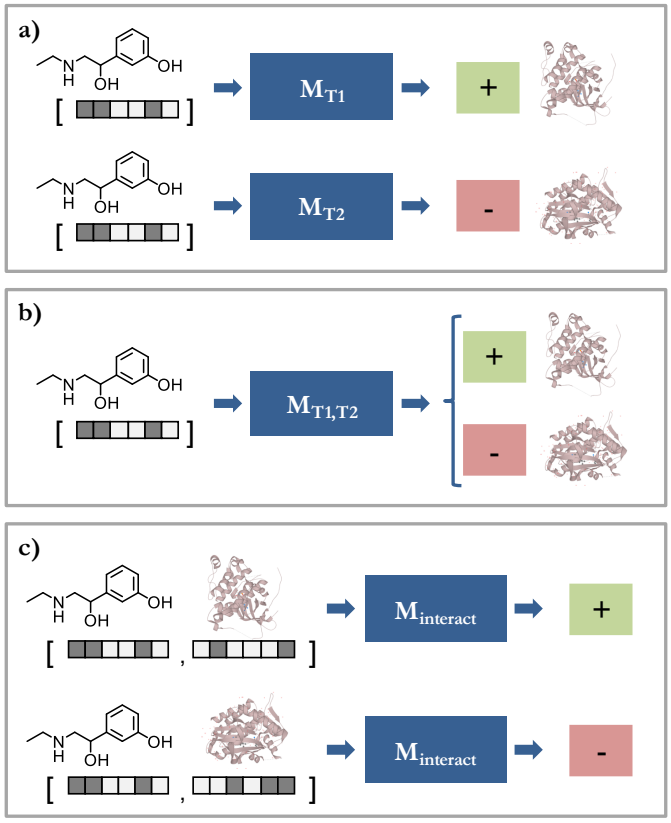


Figure 12: Modeling strategies for activity prediction. Three modeling strategies are illustrated for the simplified problem of predicting compound activity against two protein targets (T1 and T2). In (a) and (b), ST and MT models, respectively, aim at discriminating between active (green) and inactive (red) compounds from their molecular representations. In (a), two models M_{T1} and M_{T2} (one per target) are used. In (b), the $M_{T1,T2}$ simultaneously models activity against targets T1 and T2. Thus, this MT model outputs two predictions per compound. In (c), the chemogenomics model $M_{interact}$ discriminates between interacting (green) and non-interacting (red) compound-target pairs, from the combined input representation.

Model interpretation

Model interpretability is a *sine qua non* for knowledge extraction and practical utility of ML models in pharmaceutical research. In fact, the importance of "explainable ML" has been recently recognized in all fields since it enables detecting scenarios in which models behave unexpectedly, and increases model acceptance by non-experts.[106] To understand model decisions, feature importance needs to be estimated in order to determine which variables are contributing most to accurate predictions. Linear models are easily interpretable but generally provide less accuracy in activity predictions.[162] In contrast, complex ML models can learn non-linear QSARs but are not easily interpretable, in particular, DNNs. Moreover, in SAR studies interpretability depends on the ML model as well as the molecular representation.[163–165]

Chapter 2

Prediction of Compound Profiling Matrices Using Machine Learning

Introduction

Compound activity has been traditionally studied on a per-target basis, but some ligands might interact with multiple targets. Thus, the prediction of activity profiles is increasingly relevant. Profiling matrices consist of a small molecule library screened across a panel of targets so that a bioactivity profile is obtained for each compound. These matrices represent a pivotal scenario for the benchmark of VS methods. In this chapter, ML approaches are developed to model large compound profiling matrices from screening experiments. In particular, a complete profiling matrix with 109,925 diverse small molecules tested against 53 targets is predicted. Similarity searching, state-of-the-art ML and DL models are generated. Moreover, single-target and multi-target learning as well as distinct molecular representations are explored.

ACS OMEGA

Article

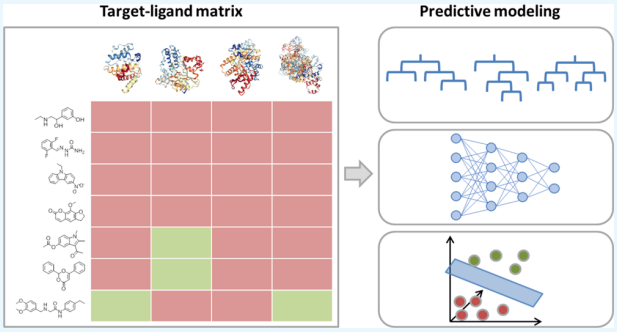
Cite This: ACS Omega 2018, 3, 4713−4723

Prediction of Compound Profiling Matrices Using Machine Learning

Raquel Rodríguez-Pérez, Tomoyuki Miyao, Swarit Jasial, Martin Vogt, and Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Endenicher Allee 19c, D-53115 Bonn, Germany

ABSTRACT: Screening of compound libraries against panels of targets yields profiling matrices. Such matrices typically contain structurally diverse screening compounds, large numbers of inactives, and small numbers of hits per assay. As such, they represent interesting and challenging test cases for computational screening and activity predictions. In this work, modeling of large compound profiling matrices was attempted that were extracted from publicly available screening data. Different machine learning methods including deep learning were compared and different prediction strategies explored. Prediction accuracy varied for assays with different numbers of active compounds, and alternative machine learning approaches often produced comparable results. Deep learning did not further increase the prediction accuracy of standard methods such as random forests or support vector machines. Target-based random forest models were prioritized and yielded successful predictions of active compounds for many assays.



1. INTRODUCTION

Machine learning methods are widely used in computational compound screening, also termed virtual screening (VS), to select limited numbers of potentially active compounds from large libraries.[1] Algorithms such as support vector machine (SVM) or random forest (RF) are among the most popular approaches for activity prediction.[2] In addition, there is increasing interest in deep learning for VS and quantitative structure−activity relationship predictions.[3−5]

Public repositories for compounds and activity data are indispensable resources for developing, evaluating, and calibrating VS methods and protocols. For small molecules and data from medicinal chemistry and biological screening, ChEMBL[6] (maintained by the European Bioinformatics Institute of the European Molecular Laboratory) and PubChem[7,8] (National Center of Biotechnology Information of the National Institutes of Health) have become primary resources, respectively. In addition, MoleculeNet has recently been introduced as a collection of curated compound activity data from diverse sources.[9] For VS benchmark calculations, known active compounds and decoys are typically assembled.[10−13] Active compounds are usually taken from medicinal chemistry sources. Evaluating VS approaches using high-throughput screening (HTS) data provides a more realistic scenario but is generally complicated by experimental variance and noise as well as natural unbalance of active and inactive compounds in HTS data sets.[14−16] Hit rates in HTS typically range from about 0.1 to 2%,[15] depending on the assays and targets, whereas most test compounds are inactive.[16] Learning from data sets of such unbalanced composition generally provides substantial challenges for deriving predictive models. Hence, predictions using HTS data are only rarely reported.[17,18] Learning from unbalanced data has been addressed in a few studies.[19−21]

In addition to state-of-the-art machine learning methods such as SVM and RF, deep neural networks (DNNs) have also been applied for activity predictions.[3−5,22−24] DNN applications sometimes report higher prediction accuracy compared with other methods. DNNs can either be trained on a per-target basis or by combining data from multiple activity classes, which are known as multitask DNNs.[23,24] Different results have been obtained by comparing the performance of single- and multitask DNNs.[23,24] A general limitation of DNN and, in particular, multitask learning is the rather limited ability to rationalize the failure of predictions.[24]

A challenge in VS going beyond learning on the basis of HTS data is the prediction of compound profiling matrices, which are obtained by screening compound collections in a panel of assays.[25−29] In these cases, the unbalance and screening data noise issues referred to above further escalate. Compounds might be active in one or more assays and inactive in others or they might be consistently inactive, yielding rather complex prediction scenarios. To our knowledge, machine learning predictions of large profiling matrices with more than just a handful of assays are yet to be reported. However, the inherent challenges of such predictions are not the only reason for their sparseness. Data unavailability is another. Although profiling matrices are frequently generated in the pharmaceutical

Received: March 12, 2018
Accepted: April 20, 2018
Published: April 30, 2018

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

Table 1. Assays and Targets[a]

assay ID	assay code	target name	organism	# active CPDs (matrix 2 training)	# active CPDs (matrix 2 test)	# active CPDs (matrix 1)
485313	A	Niemann-pick C1 protein precursor	*Homo sapiens*	3103	3142	395
485314	B	DNA polymerase β	*Homo sapiens*	1325	1326	125
485341	C	β-lactamase	*Escherichia coli*	458	478	420
485349	D	serine-protein kinase ATM isoform 1	*Homo sapiens*	191	175	118
485367	E	ATP-dependent phosphofructokinase	*Trypanosoma brucei brucei*	152	138	103
504466	F	ATPase family AAA domain-containing protein 5	*Homo sapiens*	1624	1586	424
588590	G	DNA polymerase iota	*Homo sapiens*	885	868	103
588591	H	DNA polymerase eta	*Homo sapiens*	1123	1129	39
624171	I	nuclear factor erythroid 2-related factor 2	*Homo sapiens*	367	391	118
624330	J	Rac GTPase-activating protein 1	*Homo sapiens*	491	536	156
1721	K	pyruvate kinase	*Leishmania mexicana*	433	425	39
1903	L	large T antigen	Simian virus 40	275	248	57
2101	M	glucocerebrosidase	*Homo sapiens*	73	58	41
2517	N	AP endonuclease 1	*Homo sapiens*	197	199	32
2528	O	Bloom syndrome protein	*Homo sapiens*	137	128	8
2662	P	histone-lysine *N*-methyltransferase MLL	*Homo sapiens*	10	15	3
2676	Q	relaxin/insulin-like family peptide receptor 1	*Homo sapiens*	215	195	223
463254	R	ubiquitin carboxyl-terminal hydrolase 2 isoform a	*Homo sapiens*	4	4	2
485297	S	Ras-related protein Rab-9A	*Homo sapiens*	3751	3810	410
488837	T	ryes absent homolog 2 isoform a	*Homo sapiens*	2	7	1
492947	U	β-2 adrenergic receptor	*Homo sapiens*	25	28	4
504327	V	histone acetyltransferase KAT2A	*Homo sapiens*	158	141	50
504329	W	nonstructural protein 1	influenza A virus	213	205	64
504339	X	lysine-specific demethylase 4A	*Homo sapiens*	4755	4757	1320
504842	Y	chaperonin-containing TCP-1 β subunit homolog	*Homo sapiens*	28	20	13
504845	Z	regulator of G-protein signaling 4	*Homo sapiens*	9	7	1
504847	AA	vitamin D3 receptor isoform VDRA	*Homo sapiens*	772	771	48
540317	AB	chromobox protein homolog 1	*Homo sapiens*	442	449	98
588579	AC	DNA polymerase kappa	*Homo sapiens*	354	362	6
588689	AD	genome polyprotein	dengue virus type 2	180	184	6
588795	AE	flap endonuclease 1	*Homo sapiens*	175	210	17
602179	AF	isocitrate dehydrogenase 1	*Homo sapiens*	75	81	28
602233	AG	phosphoglycerate kinase	*Trypanosoma brucei brucei*	28	40	1
602310	AH	DNA dC->dU-editing enzyme APOBEC-3G	*Homo sapiens*	60	66	11
602313	AI	DNA dC->dU-editing enzyme APOBEC-3F isoform a	*Homo sapiens*	202	183	28
602332	AJ	heat shock 70 kDa protein 5	*Homo sapiens*	15	15	6
624170	AK	glutaminase kidney isoform	*Homo sapiens*	162	186	65
624172	AL	glucagon-like peptide 1 receptor	*Homo sapiens*	7	7	2
624173	AM	hypothetical protein	*Trypanosoma brucei brucei*	136	141	32
624202	AN	breast cancer type 1 susceptibility protein	*Homo sapiens*	1469	1484	275
651644	AO	viral protein r	human immunodeficiency virus 1	208	209	74
651768	AP	Werner syndrome ATP-dependent helicase	*Homo sapiens*	278	325	5
652106	AQ	α-synuclein	*Homo sapiens*	111	102	57
720504	AR	serine/threonine-protein kinase PLK1	*Homo sapiens*	3357	3308	662
720542	AS	apical membrane antigen 1	*Plasmodium falciparum*	93	98	25
720707	AT	Rap guanine nucleotide exchange factor 3	*Homo sapiens*	50	62	3
720711	AU	Rap guanine nucleotide exchange factor 4	*Homo sapiens*	59	68	16
743255	AV	ubiquitin carboxyl-terminal hydrolase 2 isoform a	*Homo sapiens*	147	149	15
743266	AW	parathyroid hormone 1 receptor	*Homo sapiens*	66	70	79
493005	AX	Tumor susceptibility gene 101 protein	*Homo sapiens*	0	0	0
504891	AY	peptidyl-prolyl cis−trans isomerase NIMA-interacting 1	*Homo sapiens*	6	5	0
504937	AZ	sphingomyelin phosphodiesterase	*Homo sapiens*	5	9	0

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

Table 1. continued

assay ID	assay code	target name	organism	# active CPDs (matrix 2 training)	# active CPDs (matrix 2 test)	# active CPDs (matrix 1)
588456	BA	thioredoxin reductase	Rattus norvegicus	1	8	0

[a]Reported are the PubChem assay IDs, codes used here, targets, and organisms, for all 53 assays. In addition, for each assay, numbers of active compounds in the matrix 2 training and test sets and in matrix 1 are reported.

industry, they are rarely disclosed. The few profiling data sets that are publicly available are essentially limited to kinase targets and partly incomplete. Thus, there is currently no sound basis for predictive modeling of profiling matrices.

In light of these limitations, we have developed a computational methodology to extract complete profiling matrices from available screening data.[30] Applying this approach, we have generated profiling matrices of different compositions including assays for a variety of targets. These matrices consist of "real life" screening data and are characterized by generally low hit rates and the presence of many consistently inactive compounds.

Prediction of compound profiling matrices is of high relevance for chemogenomics research, which ultimately aims at accounting for all possible small molecule−target interactions. For all practical purposes, reaching this goal will essentially be infeasible. Accordingly, there is a high level of interest in computational approaches that are capable of complementing profiling experiments with reliable ligand−target predictions. Moreover, profiling matrices also represent excellent model systems for HTS campaigns using a given compound deck. If experimental matrices are available, predicting the outcome of HTS runs against different targets can be attempted under realistic conditions. This provides much more informative estimates of computational screening performance than artificial benchmark settings that are typically used. In drug discovery, the prediction of HTS data has long been and continues to be a topical issue. For example, because the capacity of (compound) "cherry-picking" from screening plates has become more widely available in the industry, computational prescreening of compound decks can be used to prioritize subsets that are most likely to yield new hits. Cycles of computational screening followed by experimental testing are implemented in iterative screening schemes, which may significantly reduce the magnitude of experimental HTS efforts.

Herein, we have applied various machine learning approaches and strategies to predict newly derived compound profiling matrices. The results are presented in the following and provide an experimentally grounded view of expected accuracy of machine learning models in predicting the outcome of screening campaigns for diverse targets.

2. RESULTS AND DISCUSSION

2.1. Profiling Matrices. Two HTS data matrices comprising the same 53 assays and targets (i.e., one assay per target) and 109 925 and 143 310 distinct compounds, respectively, were used for machine learning and VS. These matrices were assembled from confirmatory assays available in the PubChem BioAssay collection[7,8] by applying our new algorithm.[30] Assays, targets, and assay codes used in the following discussion are reported in Table 1. The density of the smaller matrix, termed matrix 1, was 100%, i.e., all possible matrix cells contained binary annotations of activity or inactivity. The number of compounds tested per assay initially ranged from 266 527 to 387 381 and 46 of the 53 assays in matrix 1 had a hit rate of less than 1%. Table 1 also shows that the number of active compounds per assay varied significantly, ranging from only a few to more than 1000. The 53 assays also included four assays without hits. For assays with only few active compounds, training of machine learning models was generally very difficult (if not impossible in some instances). However, if all test compounds were predicted to be inactive in such cases, satisfactory results would still be obtained (i.e., only very few actives would be missed), despite intrinsic limitations of model building.

A second matrix was generated by slightly reducing the density in favor of larger compound numbers.[30] From this matrix, all compounds contained in matrix 1 were removed, yielding matrix 2. The density of matrix 2 was 96%. Matrix 1 and matrix 2 contained 105 475 (96.0%) and 110 218 (76.9%) compounds, respectively, which were consistently inactive in all assays. In matrix 1, 3639 (3.3%) of the test compounds had single- and 811 (0.7%) had multitarget activity. For matrix 2, the corresponding numbers of active compounds were 19 069 (13.3%) and 14 023 (9.8%), respectively. Hence, the composition of these matrices was highly unbalanced and dominated by consistently inactive compounds. Overall, only 0.1 and 0.8% of the cells in matrix 1 and 2, respectively, contained activity annotations. Matrix composition is summarized in Table 2. In matrix 1, the number of active compounds per assay ranged from 0 to 1320, with a mean and median value of 110 and 32, respectively. In matrix 2, it ranged from 0 to 9512, with a mean and median value of 1077 and 348, respectively. Figure 1 shows exemplary active compounds from matrix 1. In Figure 2, intra- and interassay similarity of active compounds is reported. The heat map reveals low mean similarity of compounds active in different assays. Furthermore, interassay and intra-assay similarity were overall comparable. Taken together, these observations indicated that it would be challenging to detect compounds sharing the same activity on the basis of similarity calculations and distinguish between compounds with different activity.

Table 2. Matrix Composition[a]

	matrix 1	matrix 2
density	100%	96.4%
# compounds (CPDs)	109 925	143 310
# assays	53	53
percentage of active cells	0.1%	0.8%
# consistently inactive CPDs	105 475 (96%)	110 218 (76.9%)
# CPDs with single-target activity	3639 (3.3%)	19 069 (13.3%)
# CPDs with multitarget activity	811 (0.7%)	14 023 (9.8%)

[a]For matrix 1 and matrix 2, the density, number of compounds and assays, percentage of cells with activity annotations (active cells), number of consistently inactive compounds, and number of compounds with single- and multitarget activity are reported.

2.2. Prediction Strategy. The primary goal was predicting the entire matrix 1 by learning from matrix 2. Predictions were

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

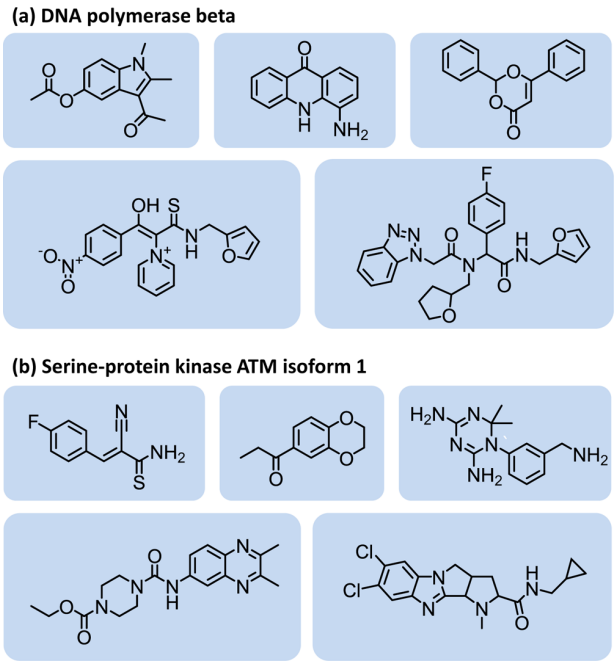


Figure 1. Exemplary active compounds. Shown are exemplary active compounds from two matrix 1 assays for (a) DNA polymerase β (assay code B) and (b) serine-protein kinase ATM isoform 1 (code D), respectively.

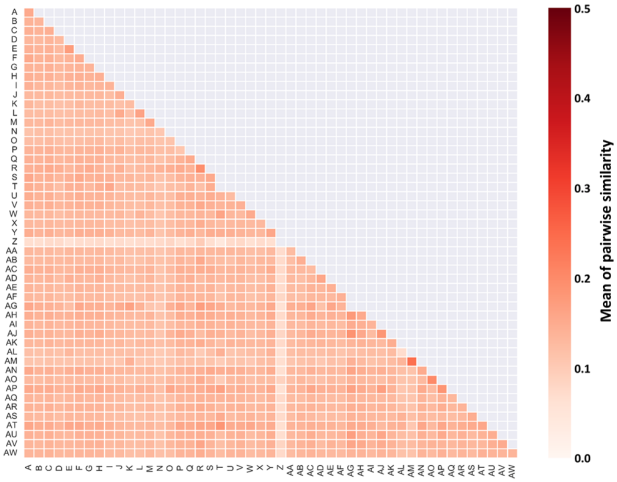


Figure 2. Pairwise Tanimoto similarity. The heat map reports mean pairwise Tanimoto similarity for active compounds from matrix 1. The extended connectivity fingerprint with bond diameter 4 (ECFP4; see Materials and Methods) was used as a molecular representation.

attempted at two levels including global predictions of active versus consistently inactive compounds as well as activity predictions for individual targets. For global models, training and test compounds with different activities were combined to yield the "active" class. Half of matrix 2 was randomly selected and used for training of global models using different methods. Global models were applied to predict active and inactive compounds for the other half of matrix 2 used as a test set as well as the entire matrix 1. Per-target models were derived in two ways: first, using half of matrix 2 and second, the entire matrix 2. The former per-target models were applied to predict the matrix 2 test set, and the complete matrix 1 and the latter models were applied to predict matrix 1. For per-target models, initial comparisons of different methods and optimization of calculation parameters were carried out for 10 assays from matrix 2 with large numbers of available training compounds (assay codes A–J in Table 1). These models were used to predict these 10 assays in the matrix 2 test set as well as in matrix 1. Further details are provided in the Materials and Methods section.

2.3. Global Models. Given that the vast majority of matrix compounds were consistently inactive in all assays, we reasoned that initial exclusion of these consistently inactive compounds followed by target-based predictions might be a promising strategy for activity prediction. Successful elimination of consistently inactive compounds would increase data balance and reduce the number of compounds to be predicted by per-target models. Therefore, global models were first built using SVM, RF, and DNN to distinguish between combined active and consistently inactive screening compounds. On the basis of test calculations (see Materials and Methods), models trained with all available data reached highest relative performance levels and the ECFP4 fingerprint was a preferred descriptor. Figure 3 shows the prediction results of the global models for

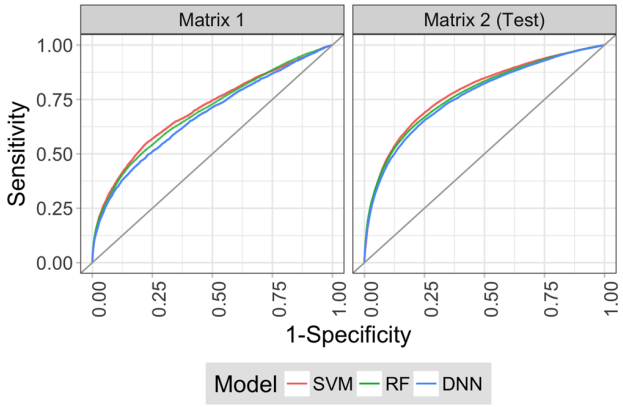


Figure 3. Receiver operating characteristic curves for global models. Receiver operating characteristic (ROC) curves are shown for SVM (red), RF (green), and DNN (blue) global models, which were trained with half of matrix 2 and used to predict the other half of matrix 2 (right) and matrix 1 (left).

the matrix 2 test set and for matrix 1. The performance of the different models was nearly identical in both cases. Although there was consistent early enrichment of active compounds, deprioritization of inactive compounds was accompanied by a substantial loss of active compounds, in particular, for matrix 1. In this case, eliminating 50% of the inactive compounds also led to a removal of 25% of the actives. For the minority class, the magnitude of this initial loss of active compounds limited the envisioned two-stage prediction approach.

2.4. Models for Assay-Based Predictions. Next, we used a subset of 10 assays with larger numbers of available active compounds (assay codes A–J in Table 1) for comparison of alternative machine learning methods and identification of best-performing models and preferred calculation conditions.

2.4.1. Method Comparison. Algorithms of different designs and complexities were systematically compared. Most of the implemented approaches resulted in single-task (per-target) models, but two multitask approaches were also included in the

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713–4723

Table 3. Area under the Curve Values for Prediction of 10 Assays of the Matrix 2 Test Set[a]

assay code	CCBM	NB	RF	SVM	single-task DNN	multitask DNN	GraphConv
A	0.85	0.84	0.91	**0.92**	0.91	0.91	0.90
B	0.77	0.79	**0.85**	**0.85**	0.82	0.82	0.83
C	0.64	0.71	**0.73**	0.72	0.69	0.67	0.72
D	0.63	**0.72**	0.69	0.65	0.67	0.62	0.64
E	0.81	0.82	**0.86**	0.84	0.84	0.85	0.85
F	0.82	0.82	**0.88**	**0.88**	0.87	0.87	0.86
G	0.73	0.79	**0.84**	**0.84**	0.81	0.79	0.82
H	0.80	0.85	**0.90**	**0.90**	0.88	0.87	0.89
I	0.80	0.85	**0.89**	**0.89**	0.88	0.85	**0.89**
J	0.84	0.87	**0.92**	**0.92**	0.91	0.86	**0.92**

[a]Reported are AUC values for prediction of 10 assays (codes A–J) using different machine learning methods. For each assay, best results are indicated in bold.

Table 4. Area under the Curve Values for Prediction of 10 Assays of Matrix 1[a]

assay code	CCBM	NB	RF	SVM	single-task DNN	multitask DNN	GraphConv
A	0.88	0.86	0.93	**0.94**	0.93	0.93	0.92
B	0.64	0.68	**0.70**	0.69	0.67	0.66	0.69
C	0.66	0.64	**0.69**	0.67	0.64	0.64	0.68
D	0.62	0.63	0.62	0.62	0.63	0.60	**0.65**
E	0.86	0.91	**0.94**	0.91	0.90	0.88	0.89
F	0.82	0.82	0.87	**0.88**	0.87	0.86	0.87
G	0.55	0.55	0.58	0.57	0.54	0.57	**0.64**
H	0.70	0.75	**0.77**	0.76	0.74	0.75	0.76
I	0.82	0.86	**0.89**	0.88	0.86	0.83	0.88
J	0.84	0.88	0.93	**0.94**	0.93	0.90	**0.94**

[a]Reported are AUC values for prediction of 10 assays (codes A–J) using different machine learning methods. For each assay, best results are indicated in bold.

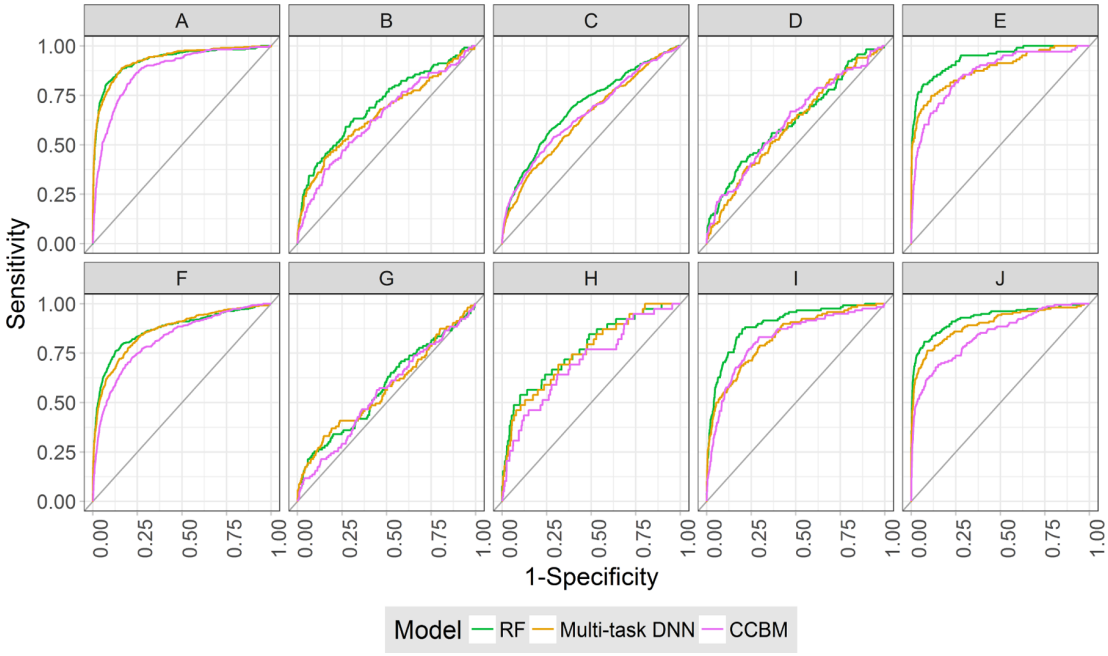


Figure 4. Per-target receiver operating characteristic curves. ROC curves are shown for target-based activity predictions with RF (green), multitask DNN (orange), and CCBM (pink) models. Curves represent 10 matrix 1 assays used for method comparisons. Codes A–J designate assays according to Table 1.

comparison. A multitask model yields probabilities of activity for compounds tested in different assays. Investigated methods included a similarity search-based approach termed the conditional correlated Bernoulli model (CCBM) to estimate rank positions of active compounds; popular machine learning approaches; such as naïve Bayes (NB) classification, SVM, and RF, single- and multitask DNN; and graph-convolutional NN (GraphConv). Test predictions were assessed by calculating

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

area under the curve (AUC) values for receiver operating characteristic (ROC) curves and recall rates for the top 1% of ranked test sets.

Initially, we tested general training conditions. For each assay, available active training compounds were combined with increasing numbers of compounds inactive in the assay and a series of models were generated with different machine learning methods and evaluated. For all methods (except GraphConv), the folded version of the extended connectivity fingerprint with bond diameter 4 (ECFP4; see Materials and Methods) was used as a descriptor. Paralleling the findings for global models, preferred training sets generally consisted of all available active and inactive training compounds. Using these training sets, different methods were compared.

Tables 3 and 4 report benchmark results for the matrix 2 test set and for matrix 1, respectively. For the matrix 2 test set, best models consistently yielded AUC values >0.7 per assay and values >0.8 for eight assays. For matrix 1, prediction accuracy was overall lower but AUC values <0.7 were only obtained for three assays. Thus, different methods yielded models with at least reasonable prediction accuracy in most cases. Interestingly, although differences in prediction accuracy were often small, RF was the overall best-performing approach, achieving top predictions for eight assays in matrix 2 and five in matrix 1. As shown in Figure 4, it also compared favorably in multitasking DNN and performed better than the CCBM similarity search control. The performance level of RF was nearly matched by SVM, followed by GraphConv. Given overall comparable prediction accuracy achieved by different machine learning methods and high RF performance across different assays, RF was selected as a representative approach for further activity predictions.

2.4.2. Alternative Molecular Representations. In the next step, RF models built using different molecular representations were compared. The results are reported in Table 5. In these calculations, ECFP4 emerged as the preferred descriptor, with nearly identical performance of its unfolded and folded (fixed length) version.

2.5. Per-Target Activity Predictions. On the basis of the comparisons above, final models for activity predictions on the 49 assays producing hits in matrix 1 were derived using RF, folded ECFP4, and all available active and inactive compounds per assay from the matrix 2 training set. As reported in Table 1, only few active training instances were available in a number of assays.

The results of activity predictions for all assays in the matrix 2 test set and in matrix 1 are reported in Figure 5. Predictions were overall superior for matrix 2 than matrix 1 (that did not share compounds with matrix 2). For matrix 2, AUC values of 0.8 or greater were achieved for 31 of 49 assays; an encouraging finding. For matrix 1, AUC values of at least 0.8 were obtained for 22 assays but there were also nine assays with low performance close to or even worse than random selection. In most cases, assays with low prediction accuracy only contained a limited number of actives (ranging from 1 to 79 compounds). As a control, matrix 1 predictions were also carried out with models trained on the entire matrix 2, shown in Figure 6. The availability of essentially twice as many active training compounds significantly improved prediction accuracy, with AUC values of 0.7 or greater obtained for 35 assays.

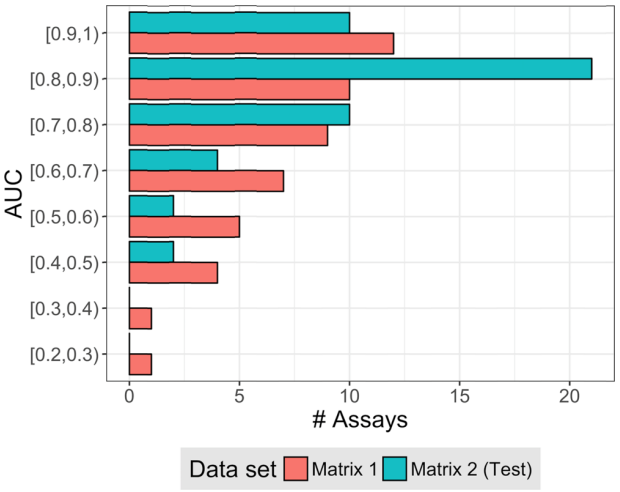


Figure 5. Area under the curve values for per-target models trained with half of matrix 2. AUC values are reported for predictions of compounds active in assays of the matrix 2 test set (blue) and matrix 1 (red).

Table 6 reports the results for predictions on the 49 assays in matrix 1 after training RF models on the entire matrix 2. Recall rates among the top 1% of the ranking ranged from 0 to 100% and varied significantly, with mean and median values of 35 and 30%, respectively. Active compounds were successfully identified for 41 of 49 assays, and 26 models achieved recall rates of at least 30%. In instances where activity predictions completely failed, only few active compounds were available (ranging from two to eight). Interestingly, for many assays, there was a notable early enrichment of active compounds. In 22 cases, the first active compound was ranked among the top three database molecules and in 30 cases, it was ranked among the top 30. Thus, per-target models yielded promising predictions in many instances.

2.6. Conclusions. In this study, we have attempted to predict compound profiling matrices extracted from raw screening data. Large numbers of assays, small numbers of active compounds, their chemical diversity, and very large number of consistently inactive compounds challenged

Table 5. Comparison of Different Molecular Representations[a]

assay code	MOE	MACCS	MOE + fold. ECFP4	nonfolded ECFP4	folded ECFP4
A	0.91	0.90	**0.93**	**0.93**	**0.93**
B	0.65	0.64	0.68	**0.70**	**0.70**
C	0.66	0.67	0.68	**0.69**	**0.69**
D	0.59	0.60	0.63	**0.65**	0.62
E	0.86	0.84	0.90	0.93	**0.94**
F	0.86	0.84	**0.87**	**0.87**	**0.87**
G	0.58	0.56	**0.60**	0.57	0.58
H	0.76	0.73	0.76	**0.77**	**0.77**
I	0.85	0.86	0.87	**0.90**	0.89
J	0.90	0.92	**0.93**	**0.93**	**0.93**

[a]Reported are AUC values for prediction of 10 assays (codes A–J) in matrix 1 using per-target RF models on the basis of different molecular representations, including 192 two-dimensional (2D) descriptors from the Molecular Operating Environment (MOE), 166 MACCS structural keys, the folded and unfolded version of ECFP4, and the combination of MOE descriptors and folded ECFP4 (MOE + fold. ECFP4). For each assay, best results are indicated in bold.

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

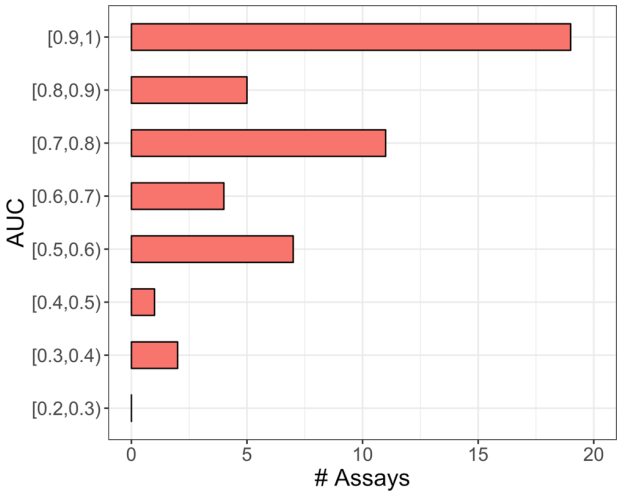


Figure 6. Area under the curve values for per-target models trained with matrix 2. AUC values are reported for predictions of compounds active in assays of matrix 1.

predictions in this case. Different machine learning methods were compared for their ability to identify active compounds across assays. Perhaps surprisingly, alternative methods often yielded comparable performance. Overall, RF emerged as a preferred approach, followed by SVM. Deep learning methods did not yield further improved prediction accuracy. In this context, we note that compound data sets used for activity predictions are still much smaller in size than many other data sets originating from life science research. In addition, compound data sets for activity prediction are also studied computationally using predefined molecular representations. Taken together, these features do not play into the strengths of deep learning in extracting patterns and feature representations from large data sets. This may explain the absence of significant performance increases through deep learning in predicting profiling matrices. Data sets originating from the life sciences that are more suitable for deep learning include, for example, images from high-content screening, data from large-scale gene expression analysis or next generation sequencing, and multipoint records from clinical trials. In these cases, performance increases through deep learning relative to other computational methods might be expected. Notably, image analysis in computer science has been one of the first applications where deep learning outperformed other machine learning approaches.

Initially, in our study, global models were designed aiming to eliminate large numbers of consistently inactive compounds. However, these models also deprioritized many active compounds, thus limiting their applicability as a first-path computational screen. By contrast, systematic activity predictions using per-target RF models yielded overall promising predictions on the basis of highly unbalanced training sets. A notable early enrichment of active compounds was frequently observed.

Compound matrices obtained from experimental screens provided a more realistic test system for machine learning than often applied benchmark settings. Under these conditions, prediction accuracy was lower than often reported for standard benchmarking exercises, as expected. Increasing complexity of machine learning methods did not scale with prediction accuracy, e.g., deep learning did not make a difference in this case. However, RF calculations yielded successful predictions for the majority of assays, indicating the ability of standard machine learning methods to identify novel active compounds under rather challenging experimental conditions. As an outlook, multitask learning should be further explored on the basis of profiling matrices for subsets of assays and we are also

Table 6. Recall of Active Compounds in the Top 1% of Ranked Matrix 1[a]

assay code	# active CPDs in matrix 1	# active CPDs in top 1%	recall (%)	rank of first active CPD
X	1320	383	29	1
S	410	209	51	2
A	395	208	53	1
F	424	161	38	1
Q	223	120	54	1
J	156	113	72	2
AN	275	80	29	1
E	103	63	61	1
AR	662	59	9	1
C	420	56	13	4
AO	74	52	70	1
W	64	49	77	1
L	57	43	75	1
AB	98	36	37	5
I	118	35	30	10
AM	32	30	94	1
M	41	28	68	2
K	39	26	67	2
AK	65	25	38	3
AQ	57	17	30	7
D	118	16	14	1
B	125	15	12	2
AA	48	15	31	1
AF	28	15	54	1
G	103	7	7	42
H	39	7	18	39
AE	17	7	41	4
AS	25	7	28	1
AV	15	6	40	11
N	32	5	16	5
Y	13	5	38	1
AI	28	5	18	19
AD	6	3	50	72
V	50	2	4	192
AJ	6	2	33	88
T	1	1	100	38
Z	1	1	100	368
AG	1	1	100	146
AH	11	1	9	32
AU	16	1	6	249
AW	79	1	1	573
O	8	0	0	6758
P	3	0	0	12 637
R	2	0	0	31 266
U	4	0	0	1805
AC	6	0	0	2012
AL	2	0	0	26 430
AP	5	0	0	1156
AT	3	0	0	36 085

[a]For each assay, the number of active compounds in matrix 1, their recall in the top 1% of the ranking, and the highest-ranked active for RF models trained with matrix 2 are reported.

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

interested in focusing predictions specifically on small numbers of compounds with multitarget activity for which a different methodological framework might be required.

3. MATERIALS AND METHODS

3.1. Matrices. A complete (100% density) assay-compound matrix (matrix 1) was generated from confirmatory assays in the PubChem BioAssay database[7] using a newly introduced algorithm.[30] PubChem compounds yielding unique SMILES representations were retained in the matrix, which contained 109 925 compounds tested against a panel of 53 different confirmatory assays. Subsequently, matrix 2 with a final density of 96% was generated using the same algorithm. Initially, a matrix 2 precursor was assembled with 95% density that contained 281 943 compounds tested in the 53 assays. From the precursor, all matrix 1 compounds were removed. In addition, 28 708 inactive compounds tested in less than 50 assays were eliminated, yielding matrix 2 with 143 310 compounds. matrix 2 was then randomly divided into training and test sets each consisting of 71 655 compounds. Zero imputation[31] was applied to missing values. Forty nine of the 53 assays produced hits, as reported in Table 1.

3.2. General Training Conditions. *3.2.1. Global Models.* Global models to distinguish between combined active and consistently inactive compounds were initially built using SVM, RF, and DNN on the basis of training sets of increasing size taken from the matrix 2 training set. A steady improvement in performance was observed with increasing training set size, consistent with earlier observations.[32] Therefore, final global models were built using the entire matrix 2 training set.

3.2.2. Per-Target Models. Initially, per-target models were trained for 10 selected assays (codes A–J) for which larger numbers of active compounds were available (Table 1). Models were built using all active training compounds and different numbers of randomly selected compounds that were inactive in each assay. First, all available inactive compounds were used. Second, the number of randomly selected inactive compounds was set to 10 and 20 times the number of active compounds, following previously established rules for composition of training sets.[32] Hence, three training sets with increasing ratio of inactive to active compounds were compared in model building.

3.3. Molecular Representations. Several descriptors were evaluated to represent compounds, including the extended connectivity fingerprints of bond diameter 4 (ECFP4)[33] and MACCS structural keys.[34] ECFP4 is a feature set fingerprint that enumerates layered atom environments and encodes them as integers using a hashing function. The feature set ("unfolded") version of ECFP4 has variable size but can be "folded" to yield a constant number of bits. A 1024 bit folded version of ECFP4 was obtained through modulo mapping. MACCS is a binary keyed fingerprint, accounting for the presence or absence of 166 predefined substructures. The OEChem toolkit[35] and inhouse Python scripts were used to generate these fingerprints. In addition, 192 numerical 2D MOE descriptors were used.[36] Among others, these descriptors included physical properties, atom and bound counts, and various topological descriptors. Furthermore, graph-based representation known as graph-convolutional networks (GraphConv) was evaluated as an alternative to conventional chemical descriptors. GraphConv is a learnable representation inspired by the Morgan circular fingerprint representing compounds as undirected graphs and employs convolutional layers to create graph-based features.[37−39] The DeepChem (version 1.3.2 dev)[40] implementation of GraphConv was used. Fingerprint similarity was quantified by calculating the Tanimoto coefficient (Tc).[41]

3.4. Machine Learning Models. Similarity searching, three state-of-the-art machine learning, and three types of DNNs were applied. For building predictive models, training compounds were represented as a feature vector $x \in \mathcal{X}$ and associated with a class label $y \in \{-1, 1\}$, encoding inactivity or activity for a given target. If the activity against all targets was predicted with a global model, y was expressed in a vector form.

3.4.1. Conditional Correlated Bernoulli Model (CCBM). CCBM is an approach for modeling the distribution of Tc values of a screening database given a reference compound.[42] For a specific target, each active compound from the matrix 2 training set was used once as the reference to search for active compounds in the test sets, i.e., matrix 2 test set and in matrix 1. Consistently inactive compounds from the matrix 2 training set were used as the database, and all active compounds present in matrix 2 test set and in matrix 1 were used as probes. A p-value representing the probability of finding a database compound with higher rank was calculated for every test compound. A nearest neighbor reference compound was determined and selected for each test compound having the highest Tc value, and the p-value corresponding to this reference compound was considered. If multiple nearest neighbors existed for a test compound, the mean p-values was taken. Finally, a ranking of test compounds was generated in the order of increasing p-values.

3.4.2. Support Vector Machine (SVM). SVM is a supervised learning algorithm aiming to identify a hyperplane H that best separates two classes using the training data projected into the feature space $\mathcal{X}$.[43] This hyperplane is defined by a weight vector $\mathbf{w}$ and a bias b so that $H = \{\mathbf{x}\cdot\mathbf{w}, \mathbf{x} + b = 0\}$ and maximizes the margin between the classes. To achieve better model generalization, slack variables can be added to permit errors of training instances falling within the margin or on the incorrect side of H. The trade-off between training errors and margin size can be controlled by the regularization hyperparameter C, which was optimized herein by 2-fold cross-validation using candidate values 0.1, 1, and 10. The preferred C values were 0.1 for 9 out of 10 models. In addition, the "kernel trick" enables projecting the training data into a higher dimensional space $\mathcal{H}$ without computing the explicit mapping of $\mathcal{X}$ into $\mathcal{H}$. Class weights were considered to preferentially penalize errors in the minority class (active compounds).[32] The Tanimoto kernel[44] was used to replace the standard scalar product.[32] SVM models were generated using SVM-light.[45]

3.4.3. Random Forest (RF). RF consists of an ensemble of decision trees built from distinct subsets of the training data with replacement, known as bootstrapping.[46] A random subset of features is considered during node splitting for the construction of trees.[47] The number of trees was set at 500, and class weights were applied. The number of randomly selected features available at each split (max_features) and the minimum number of samples required to reach a leaf node (min_samples_leaf) were optimized via 2-fold cross-validation. Candidate values for max_features were the square root, the logarithm to base 2, or the total number of features; for min_samples_leaf, candidate values were 1, 5, and 10. RF calculations were carried out with scikit-learn.[48] The minimum number of samples for a leaf node was set to 5 for half of the assays and to 10 for the other half and the maximum number of

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

features to 10 and 32, respectively. No preferred parameter combination was identified.

3.4.4. Naïve Bayes (NB). NB uses Bayes' theorem to predict the probability of a compound x to be active assuming feature independence[49,50]

$$P(\text{active}|x) = \frac{P(x|\text{active})\cdot P(\text{active})}{P(x)}$$

For binary descriptors, the Bernoulli NB implementation of scikit-learn was used.[48]

3.4.5. Deep Feed Forward Neural Network (DNN). DNN classifier approximates a function that maps an input value x to a class y, $y = f(x; w)$, and learns the value of parameters w to achieve the best approximation.[51] DNN consists of different layers with a number of neurons: an input layer, at least two hidden layers, and an output layer.[52] Each hidden or output neuron assigns weights to the inputs, adds these weights, and passes the sum through a nonlinear function or activation function

$$y_k = f\left(\sum_j w_{kj}x_j + b_k\right)$$

where y is the output of neuron k, f is the activation function, x is the input variable (activation neuron in the previous layer), w is the weights connecting neuron k with x_j, and b_k is the bias. The summation includes all of the neurons adding connections to k.[53] Accordingly, each input is modified by a unique set of weights and biases. During the training phase, weights and biases are modified to obtain the correct output y, which is facilitated by following the gradient of the cost function (gradient decent) and efficiently calculated using back-propagation.[52] Training is generally performed using subsets of data, and the weights and biases are updated accordingly. Single-task DNNs (with one DNN per assay) and a multitask DNN (i.e., a single DNN for predicting all active compounds) were investigated. For the multitask DNN, the matrix containing the activity profiles for training compounds was fed into the network as the set of desired outputs y and the output layer consisted of multiple nodes equaling the number of assays. Implementations were based on tensorflow[54] and keras.[55]

Following previously formulated guidelines,[4,24,56] hyperparameters were either set to constant values or optimized by internal validation using 80 vs 20% data splits. For DNN, tested values for the learning rate (LR) were 0.01 and 0.001 and for the drop-out rate (DO), tested values were 25 and 50%. Investigated network architectures included [2000, 100], [2000, 1000], [500, 500, 500], [2000, 1000, 100], and [2000, 1000, 500, 100]. Therefore, both pyramidal and rectangular architectures were considered during hyperparameter optimization. Stochastic gradient descent was chosen as the optimizer, 128 as the batch size, and the "rectified linear unit" (ReLU) as the activation function. Output nodes were "softmax" for the single-task and "sigmoid" for the multitask DNNs. Different weights were also applied to the data according to the ratio of the number of active to inactive compounds to put more emphasis on actives. Finally, the maximum number of "epochs" was set to 100 for internal validation and 500 for the final model building.

For single-task DNN, two combinations of hyperparameters were preferentially selected including an optimum LR of 0.001, DO of 50%, and architecture [2000, 1000], as well as LR was of 0.01, DO of 25%, and architecture [2000, 100]. Multitask models require a single combination of optimized hyperparameters. Therefore, the median of AUC for all assays was used as a metric for multitask DNN hyperparameter optimization. The maximum value was obtained with a pyramidal architecture of two layers ([2000, 1000]), LR of 0.001, and DO of 25%.

3.4.6. Graph-Convolutional Neural Networks (GraphConv). As mentioned above, GraphConv is based on features or descriptors with learnable parameters from a 2D molecular graph. Initially, a set of atom features, such as atom type or valence, and a neighbor list is obtained for every atom. Neighbor information is assigned to each atom by summing up the neighbors' features. The learnable parameters include the weight matrices and biases used for posterior transformations. The same weight matrices and bias vectors are used in one layer depending on the degrees of atoms. After updating atom features, the pooling layer uses an activation function to generate a new set of feature values, which is the output vector in one layer. This procedure is repeated several times, and all of the outputs are summed up to obtain the final representation of the compound.[5] Finally, this representation is the input of a fully connected DNN. Therefore, in this approach, feature extraction and model building are combined into one trainable module.[38] In our study, GraphConv models were carried out with DeepChem (version 1.3.2 dev),[40] which implemented a modified architecture of GraphConv. The pooling operator is max pool on an atom that returns the maximum activation across the atom and the atom's neighbors without introducing additional parameters. Instead of summing several layers' outputs, a graph gather layer is introduced. This layer sums all feature vectors for all atoms to obtain the final representation of a compound.

For GraphConv, internal validation (80−20%) was also applied and the number of epochs was set to 50 and the batch size to 256. The DO value was set to 25%. The numbers of output features in hidden graph-convolutional layers were [64], [64, 64], [64, 128, 64], [32, 32, 32, 32], and [64, 64, 64, 64]; for the dense layer dimension, which precedes the gather layer, they were 128 and 256; and for LR they were 0.01 and 0.001. Moreover, batch normalization, Adam optimizer, and ReLU were considered except for the gather (tanh), as the default settings in DeepChem. A single combination of hyperparameters was determined on the basis of the median value of AUCs for the 10 assays, as described for multitask DNN. The preferred architecture had three hidden convolutional layers with [64, 128, 64] neurons, 256 neurons in the dense layer, and an LR of 0.001.

AUTHOR INFORMATION

Corresponding Author

*E-mail: bajorath@bit.uni-bonn.de. Phone: 49-228-7369-100.

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Author Contributions

The study was carried out and the manuscript written with contributions of all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare no competing financial interest.

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

■ ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.-P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 676434, "Big Data in Chemistry" ("BIGCHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) are responsible for any use that may be made of the information it contains. T.M. is a JSPS Overseas Research Fellow of Japan Society for the Promotion of Science. We acknowledge the OpenEye Scientific Software, Inc., for providing a free academic license of the OpenEye toolkit.

■ REFERENCES

(1) Kim, S. Getting the Most out of PubChem for Virtual Screening. *Expert Opin. Drug Discovery* **2016**, *11*, 843−855.
(2) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Cheminformatics: Quo Vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413−1437.
(3) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3−14.
(4) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.
(5) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discovery Today*, in press, **2018**. 10.1016/j.drudis.2018.01.039.
(6) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papatados, G.; Santos, R.; Overington, J. P. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, D1083−1090.
(7) Wang, Y.; Bryant, S. H.; Cheng, T.; Wang, J.; Gindulyte, A.; Shoemaker, B. A.; Thiessen, P. A.; He, S.; Zhang, J. PubChem BioAssay: 2017 Update. *Nucleic Acids Res.* **2017**, *45*, D955−D963.
(8) Wang, Y.; Cheng, T.; Bryant, S. H. PubChem BioAssay: A Decade's Development toward Open High-Throughput Screening Data Sharing. *SLAS Discovery* **2017**, *22*, 655−666.
(9) 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. 2017, arXiv:1703.00564. arXiv.org e-Print archive. https://arxiv.org/abs/1703.00564.
(10) Chen, B.; Harrison, R. F.; Papadatos, G.; Willett, P.; Wood, D. J.; Lewell, X. Q.; Greenidge, P.; Stiefl, N. Evaluation of Machine-Learning Methods for Ligand-Based Virtual Screening. *J. Comput.-Aided Mol. Des.* **2007**, *21*, 53−62.
(11) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49*, 169−184.
(12) Tiikkainen, P.; Markt, P.; Wolber, G.; Kirchmair, J.; Distinto, S.; Poso, A.; Kallioniemi, O. Critical Comparison of Virtual Screening Methods against the MUV Data Set. *J. Chem. Inf. Model.* **2009**, *49*, 2168−2178.
(13) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594.
(14) Xie, X.-Q. S. Exploiting PubChem for Virtual Screening. *Expert Opin. Drug Discovery* **2010**, *5*, 1205−1220.
(15) Lipinski, C. A. Overview of Hit to Lead: The Medicinal Chemist's Role from HTS Retest to Lead Optimization Hand Off. In *Lead-Seeking Approaches*; Hayward, M. M., Ed.; Springer: New York, 2010; pp 1−24.
(16) Spencer, R. W. High-Throughput Screening of Historic Collections: Observations on File Size, Biological Targets, and File Diversity. *Biotechnol. Bioeng.* **1998**, *61*, 61−67.
(17) Hao, M.; Wang, Y. L.; Bryant, S. H. An Efficient Algorithm Coupled with Synthetic Minority Over-Sampling Technique to Classify Imbalanced PubChem BioAssay Data. *Anal. Chim. Acta* **2014**, *806*, 117−27.
(18) Han, L.; Wang, Y.; Bryant, S. H. Developing and Validating Predictive Decision Tree Models from Mining Chemical Structural Fingerprints and High-throughput Screening Data in PubChem. *BMC Bioinf.* **2008**, *9*, No. e401.
(19) Tang, Y.; Zhang, Y.; Chawla, N. V.; Krasser, S. SVM Modelling for Highly Imbalanced Classification. *IEEE Trans. Syst. Man Cybern., Part B Cybern.* **2009**, *39*, 281−288.
(20) Li, Q.; Wang, Y.; Bryant, S. H. A Novel Method for Mining Highly Imbalanced High-Throughput Screening Data in PubChem. *Bioinformatics* **2009**, *25*, 3310−3316.
(21) Zakharov, A. V.; Peach, M. L.; Sitzmann, M.; Nicklaus, M. C. QSAR Modeling of Imbalanced High-Throughput Screening Data in PubChem. *J. Chem. Inf. Model.* **2014**, *54*, 705−712.
(22) 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. Cheminf.* **2017**, *9*, No. e45.
(23) Erhan, C.; L'Heureux, P.-J.; Yue, S. Y.; Bengio, Y. Collaborative Filtering on a Family of Biological Targets. *J. Chem. Inf. Model.* **2006**, *46*, 626−635.
(24) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.
(25) Fabian, M. A.; Biggs, W. H., 3rd; Treiber, D. K.; Atteridge, C. E.; Azimioara, M. D.; Benedetti, M. G.; Carter, T. A.; Ciceri, P.; Edeen, P. T.; Floyd, M.; Ford, J. M.; Galvin, M.; Gerlach, J. L.; Grotzfeld, R. M.; Herrgard, S.; Insko, D. E.; Insko, M. A.; Lai, A. G.; Lélias, J. M.; Mehta, S. A.; Milanov, Z. V.; Velasco, A. M.; Wodicka, L. M.; Patel, H. K.; Zarrinkar, P. P.; Lockhart, D. J. A Small Molecule-Kinase Interaction Map for Clinical Kinase Inhibitors. *Nat. Biotechnol.* **2005**, *23*, 329−336.
(26) Anastassiadis, T.; Deacon, S. W.; Devarajan, K.; Ma, H.; Peterson, J. R. Comprehensive Assay of Kinase Catalytic Activity Reveals Features of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2011**, *29*, 1039−1045.
(27) Metz, J. T.; Johnson, E. F.; Soni, N. B.; Merta, P. J.; Kifle, L.; Hajduk, P. J. Navigating the Kinome. *Nat. Chem. Biol.* **2011**, *7*, 200−202.
(28) Elkins, J. M.; Fedele, V.; Szklarz, M.; Azeez, K. R. A.; Salah, E.; Mikolajczyk, J.; Romanov, S.; Sepetov, N.; Huang, X. P.; Roth, B. L.; Zen, A. H.; Fourches, D.; Muratov, E.; Tropsha, A.; Morris, J.; Teicher, B. A.; Kunkel, M.; Polley, E.; Lackey, K. E.; Atkinson, F. L.; Overington, J. P.; Bamborough, P.; Müller, S.; Price, D. J.; Willson, T. M.; Drewry, D. H.; Knapp, S.; Zuercher, W. J. Comprehensive Characterization of the Published Kinase Inhibitor Set. *Nat. Biotechnol.* **2016**, *34*, 95−103.
(29) Klaeger, S.; Heinzlmeir, S.; Wilhelm, M.; Polzer, H.; Vick, B.; Koenig, P. A.; Reinecke, M.; Ruprecht, B.; Petzoldt, S.; Meng, C.; Zecha, J.; Reiter, K.; Qiao, H.; Helm, D.; Koch, H.; Schoof, M.; Canevari, G.; Casale, E.; Depaolini, S. R.; Feuchtinger, A.; Wu, Z.; Schmidt, T.; Rueckert, L.; Becker, W.; Huenges, J.; Garz, A. K.; Gohlke, B. O.; Zolg, D. P.; Kayser, G.; Vooder, T.; Preissner, R.; Hahne, H.; Tõnisson, N.; Kramer, K.; Götze, K.; Bassermann, F.; Schlegl, J.; Ehrlich, H. C.; Aiche, S.; Walch, A.; Greif, P. A.; Schneider, S.; Felder, E. R.; Ruland, J.; Médard, G.; Jeremias, I.; Spiekermann, K.; Kuster, B. The Target Landscape of Clinical Kinase Inhibitors. *Science* **2017**, *358*, No. eaan4368.
(30) Vogt, M.; Jasial, S.; Bajorath, J. Extracting Compound Profiling Matrices from Screening Data. *ACS Omega* **2018**, DOI: 10.1021/acsomega.8b00461, in press.
(31) Tanrikulu, Y.; Kondru, R.; Schneider, G.; So, W. V.; Bitter, H. Missing Value Estimation for Compound-Target Activity Data. *Mol. Inf.* **2010**, *29*, 678−684.
(32) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.
(33) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(34) *MACCS Structural Keys*; Accelrys: San Diego, CA, 2011.
(35) *OEChem TK*, version 2.0.0; OpenEye Scientific Software: Santa Fe, NM, 2015.
(36) *Molecular Operating Environment (MOE)*; Chemical Computing Group ULC: Montreal, QC, Canada, 2018.
(37) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures − a Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.
(38) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gomez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. In *Convolutional Networks on Graph for Learning Molecular Fingerprints*, Advances in Neural Information Processing Systems, 2015; Vol. *28*, pp 2224−2232.
(39) Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Cent. Sci.* **2017**, *3*, 283−293.
(40) *DeepChem: Deep-Learning Models for Drug Discovery and Quantum Chemistry*, 2017. https://github.com/deepchem/deepchem (accessed Jan 17, 2018).
(41) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(42) Vogt, M.; Bajorath, J. Introduction of the Conditional Correlated Bernoulli Model of Similarity Value Distributions and its Application to the Prospective Prediction of Fingerprint Search Performance. *J. Chem. Inf. Model.* **2011**, *51*, 2496−2506.
(43) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.
(44) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Netw.* **2005**, *18*, 1093−1110.
(45) Joachims, T. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods: Support Vector Learning*; Schölkopf, B.; Burges, C. J. C.; Smola, A. J., Eds.; MIT Press: Cambridge, 1998; pp 169−184.
(46) Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **1979**, *7*, 1−26.
(47) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(48) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(49) Alpaydin, E. *Introduction to Machine Learning*, 2nd ed.; MIT Press: Cambridge, 2010.
(50) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*, 2nd ed.; Wiley-Interscience: New York, 2000.
(51) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, 2016.
(52) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015.
(53) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.
(54) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M., Yu, Y.; Zheng, X. In *TensorFlow: A System for Large-Scale Machine Learning*, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, Nov 2−4, 2016; USENIX Association, 2016.
(55) Chollet, F. *Keras*, version 2.1.3, 2015. https://github.com/keras-team/keras (accessed Jan 17, 2018).
(56) Koutsoukas, A.; Monaghan, K. J.; Xiaoli, L.; Huan, J. Deep-Learning: Investigating Deep Neural Networks Hyper-parameters and Comparison of Performance to Shallow Methods for Modelling Bioactivity Data. *J. Cheminf.* **2017**, *9*, No. e42.

DOI: 10.1021/acsomega.8b00462
ACS Omega 2018, 3, 4713−4723

Summary

Compound profiling matrices from "real life" screening data were modeled using methods of increasing complexity, from the traditional similarity searching to graph convolutional networks. Some methods resulted in ST and others in MT configurations, allowing the individual or simultaneous modeling of distinct biological activities, respectively. Even though similarity searching provided the baseline approach, ML models of rather different complexity often provided similar results, without a clearly superior method in terms of predictive performance. Interestingly, MT-DL did not provide a substantial performance benefit over standard ML algorithms such as RF or SVM. Therefore, final predictions were obtained with per-target RF models, which are easier to interpret and train than DNNs. For many assays, accurate ML-based predictions were achieved even with structurally diverse hits, experimental variance and noise as well as large class imbalance. Taken together, the results provided an estimation of expected performance and preferred strategies when modeling profiling matrices coming from screening data.

In the following chapter, MT learning strategies including MT-DNNs are compared for another relevant application of predictive pharmaceutical modeling.

Chapter 3

Multitask Machine Learning for Classifying Highly and Weakly Potent Kinase Inhibitors

Introduction

In the previous chapter, MT-DL models did not produce any significant benefit for the prediction of compound activity against multiple unrelated targets in screening data. In this chapter, the potential of MT learning is assessed for related prediction tasks and the discrimination between highly and weakly potent kinase inhibitors. The data set includes 19,030 inhibitors with activity against 103 kinases. Due to the lack of implementations of traditional MT-ML allowing for missing annotations, MT-DNNs are compared to ST-ML, combining the benefits of MT and DL. Herein, MT-DNN performance is rationalized by comparing it to other standard ML algorithms, including individual and simultaneous modeling of biological targets.

ACS OMEGA

Article

Cite This: ACS Omega 2019, 4, 4367−4375

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

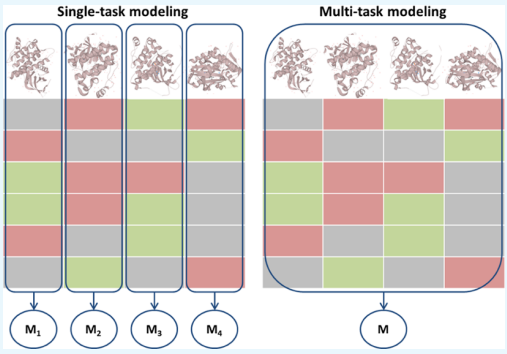
Multitask Machine Learning for Classifying Highly and Weakly Potent Kinase Inhibitors

Raquel Rodríguez-Pérez[†,‡] and Jürgen Bajorath*[,†]

[†]Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Endenicher Allee 19c, D-53115 Bonn, Germany

[‡]Department of Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, Birkendorfer Str. 65, 88397 Biberach/Riß, Germany

ABSTRACT: Compound activity prediction is a major application of machine learning (ML) in pharmaceutical research. Conventional single-task (ST) learning aims to predict active compounds for a given target. In addition, multitask (MT) learning attempts to simultaneously predict active compounds for multiple targets. The underlying rationale of MT learning is to guide and further improve modeling by exploring and exploiting related prediction tasks. For MT learning, deep neural networks (DNNs) are often used, establishing a link between MT and deep learning. In this work, ST and MT strategies for ML methods including DNN were compared in the systematic prediction of highly potent and weakly potent protein kinase inhibitors. A total of 19 030 inhibitors with activity against 103 human kinases were used for modeling. Given its composition, the data set provided many related prediction tasks. DNN, support vector machine, and random forest ST and MT models were derived and compared. Clear trends emerged. Regardless of the method, MT learning consistently outperformed ST modeling. Overall MT-DNNs achieved the highest prediction accuracy, but advantages over other MT-ML methods were only marginal. Furthermore, feature weights were extracted from models to evaluate correlation between different prediction tasks.



1. INTRODUCTION

Machine learning (ML) is widely used for the prediction of compound properties including bioactivity.[1] Within the ML spectrum, deep learning (DL) has experienced increasing interest in many fields including drug design and chemoinformatics.[2−5] Notably, the multitask (MT) learning paradigm is one of the major reasons for high expectations associated with DL in pharmaceutical research.[6,7] Activity prediction can be understood as a multilabel classification problem taking into consideration that compounds might be active against multiple targets. For example, predicting inhibitors of related targets such as protein kinases that may have single- or multitarget activity represents a multilabel classification scenario and involves related prediction tasks.

MT learning represents a generalization of the multilabel case. The characteristic feature of MT learning is that all tasks are simultaneously modeled.[8] Some ML methods are inherently suitable for MT modeling or permit *algorithmic adaptation*, whereas others require the initial derivation of binary models, which is often referred to as *problem transformation*.[9] The most common form of problem transformation is the *binary relevance* approach, which requires the generation of a single model per class (or target), that is, a ST model. There are other transformation strategies such as the *one-vs-one* approach (i.e., one classifier is generated for every combination of two classes), the *classifier chain* (i.e., ST models are iteratively built and the model output is used as an input for the next generation),[10] or the *label powerset* (where each unique combination of labels is considered a different class).

Deep neural networks (DNNs) are able to simultaneously model compound activity against multiple targets through the use of multiple output neurons. This architecture is known as MT-DNN. In general, DL classifiers are capable of generating different levels of abstraction from prediction tasks to derive complex mathematical functions taking task correlation into account. Accordingly, these complex architectures are thought to be suitable for modeling of large data sets and expected to capture more elaborate patterns or properties than standard ML techniques.[4] However, although a few recent studies have reported superior performance of DL compared to other methods in activity or target prediction,[3,6,7] others have found no consistent advantage of ST- and MT-DNN models over other ML approaches such as random forest (RF) or support vector machine (SVM) algorithms.[11−13] It has also been attempted to better understand which prediction conditions

Received: February 1, 2019
Accepted: February 19, 2019
Published: February 27, 2019

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

might generally favor MT-DNNs.[6,11] For example, Xu et al. have shown that MT-DNNs were superior to ST-DNNs if test compounds had similar structures to training compounds from other tasks as well as correlated activities.[11] On the other hand, the presence of uncorrelated tasks (e.g., targets for which compound overlap or similarity is limited) negatively influences MT modeling.[11] Moreover, the density of compound−target matrices was found to influence the relative performance of MT-DNN and ST-RF models; MT-DNN was only superior to ST-RF when activity annotations were very sparsely distributed over the matrix.[13] MT learning is not only limited to DNNs but also feasible for other ML methods such as RF.[14] However, most currently available MT-ML implementations cannot handle missing data labels, which complicate applications of MT modeling.

On the basis of currently available results, no firm conclusions can be drawn concerning the potential superiority of DL compared to other ML approaches in activity prediction. More studies will be required. In this work, ST- and MT-DNNs have been compared to other ST- and MT-ML approaches in predicting inhibitors for a large panel of kinases. To these ends, different MT learning strategies were implemented and compared.

2. RESULTS AND DISCUSSION

Different ST- and MT-ML (RF, SVM, and DNN) approaches were evaluated for predicting highly and weakly potent inhibitors of a panel of 103 kinases. In addition, model-based feature weight (FW) correlation between prediction tasks was explored and related to MT modeling performance.

2.1. Inhibitor−Kinase Interactions. Figure 1 shows the distributions of highly and weakly potent inhibitors per kinase.

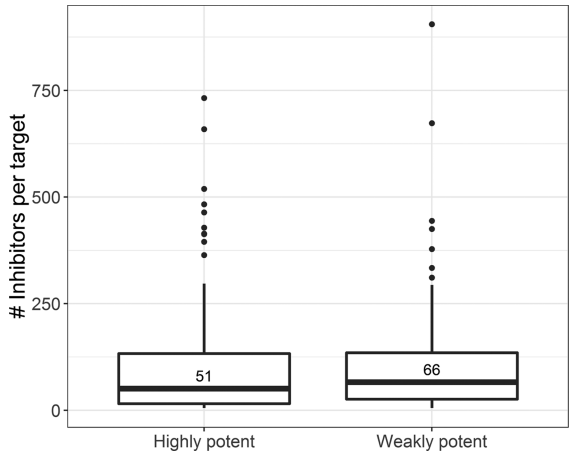


Figure 1. Distribution of highly and weakly potent inhibitors per kinase. Box plots report the distributions of the number of highly potent (left) and weakly potent (right) inhibitors per kinase. Horizontal lines indicate distribution median values (highly potent: 51; weakly potent: 66).

Compound numbers varied significantly, with median values 51 (highly potent) and 66 (weakly potent), respectively. Overall, 50% of the inhibitors were highly potent against a single kinase, 46% were weakly potent against one or more kinases, and 4% were highly potent against multiple kinases.

2.2. Training and Test Sets. For ST and MT learning, training and test sets were generated in two different ways. First, inhibitors were randomly separated into training and test sets, referred to as compound-based splitting. Second, inhibitors were organized into a series of structural analogs and singletons. Then, training and test sets were assembled by separating complete analog series (ASs) and singletons. This approach was termed AS-based splitting. It ensured that training and test sets did not contain structural analogs, which provided more challenging prediction conditions than randomly generated training and test sets. The alternative approaches are illustrated in Figure 2.

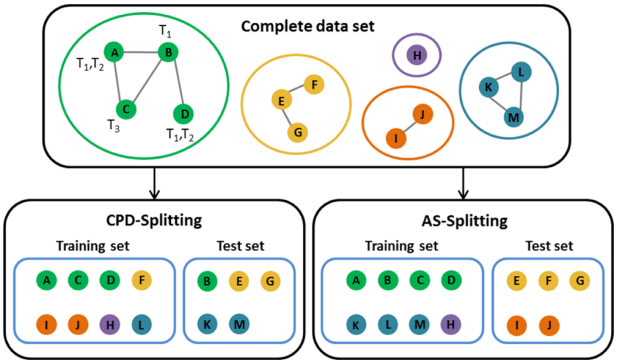


Figure 2. Strategies for generating training and test sets. Compounds (CPDs) from the complete data set (labeled A−M) are divided into ASs. Edges between compounds indicate MMP relationships, and each separate cluster represents a color-coded AS (including singletons). Each compound has one or more recorded target annotations (T_x), as shown for the green cluster. Two strategies are applied to separate compounds into training and test sets including compound-based splitting (CPD-splitting) and AS-based splitting (AS-splitting). In CPD-splitting, individual compounds are assigned to the training or test set, regardless of AS membership. In AS-splitting, complete ASs are included in the training or test set.

2.3. MT Learning Strategies. For MT learning, an algorithmic DNN implementation (MT_Algorithm) was used. Because no algorithmic MT implementations capable of treating missing labels were available for RF and SVM, an algorithm-independent MT learning strategy was devised. This approach was termed MT_Identity. For each inhibitor−kinase pair, a feature vector was generated by concatenating a compound fingerprint and target identity vector encoding the kinase, as illustrated in Figure 3. Accordingly, feature vectors represented individual interactions and inclusion of target information supported MT learning. To predict highly potent inhibitors, a global binary model was trained on the basis of feature vectors representing strong or weak inhibitor−kinase interactions. Test instances were also encoded as feature vectors containing compound and target information and inhibitors were predicted for designated targets. Application of the MT_Identity strategy enabled MT-RF and MT-SVM modeling.

2.4. Global Performance of DNN Models. Initially, ST-DNN and different MT-DNN models were generated, and their predictions were compared. ST models were separately built on the basis of inhibitors available for each kinase (resulting in a model per kinase). As compound representations, two fingerprints of different designs were used, including MACCS and the extended connectivity fingerprint with bond diameter 4 (ECFP4) (see Materials and Methods).

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

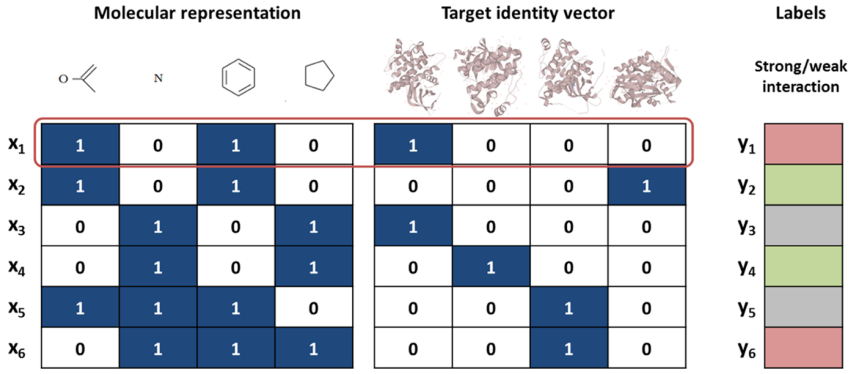


Figure 3. Feature vectors for the MT_Identity strategy. Feature vectors x_n with labels y_n represent inhibitor−kinase interactions (x_1 is marked in red). A feature vector combines a compound fingerprint and target identity vector encoding the kinase. Label coloring characterizes interactions (green: highly potent inhibitor/strong interaction, red: weakly potent inhibitor/weak interaction, gray: missing activity annotation).

Figure 4 shows Matthew's correlation coefficient (MCC) values for the different DNN models. Predictions are reported over all inhibitor−kinase interactions (entire matrix) and on a per-target basis. MT-DNN models were consistently superior to ST-DNN models, regardless of the molecular representations and splitting strategies used. Assessing predictions on a per-target basis followed by averaging revealed overall more variations than assessing predictions on the basis of the entire matrix and provided more details. On a per-target basis, AS-based predictions were generally of lower accuracy than compound-based predictions, as anticipated. Interestingly, the performance of MT_Algorithm and MT_Identity models was overall comparable, lending credence to the algorithm-independent MT_Identity strategy. Differences only became apparent when AS-based predictions were evaluated on a per-target basis. Here, models built using the algorithmic MT-DNN implementation were on average slightly more accurate than the MT_Identity models. Given the higher resolution of per-target than matrix-based evaluation of predictions, models generated using different ML methods were subsequently compared on a per-target basis.

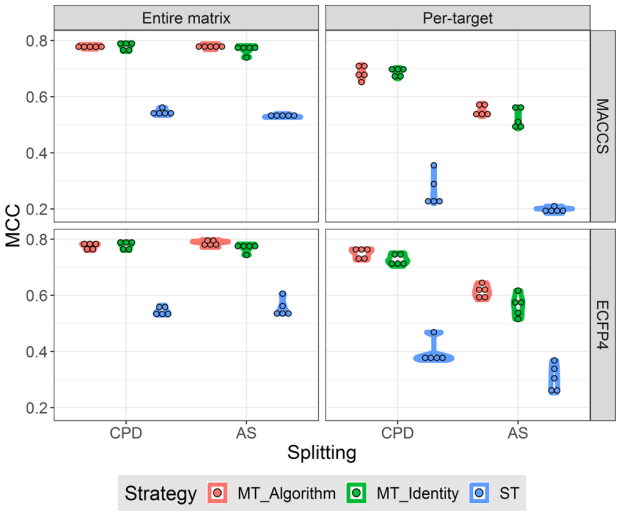


Figure 4. Global performance of DNN models. Reported are mean MCC values per trial for the entire matrix (left) and per-target predictions (right). Each dot represents an individual trial. Prediction strategies include algorithmic MT-DNN (salmon), MT-DNN based on the identity vector (green), and ST-DNN (blue). Results are reported for the MACCS (top) and ECFP4 (bottom) fingerprints and the CPD- and AS-splitting strategies (x-axis) according to Figure 2.

2.5. Method Comparison. *2.5.1. ST-RF versus MT-DNN Models.* Next, ST-RF and MT-DNN (MT_Algorithm) models were compared, which mark opposite ends of the computational complexity spectrum of ML methods investigated herein. In previous studies on ligand−target matrices from screening experiments, ST-RF models yielded overall accurate activity predictions[12,13] that were only further improved by MT-DNN if training data were very sparse.[13] However, different from predictions of screening data matrices containing unrelated targets and active as well as inactive compounds, the kinase data set investigated here provided related prediction tasks and thus different modeling conditions.

Figure 5 reports distributions of balanced accuracy (BA) and MCC values for predictions using ST-RF and MT-DNN models evaluated on a per-target basis. For kinase inhibitors, MT-DNN models produced consistently more accurate predictions than ST-RF models, which was attributable to the presence of related prediction tasks. For both performance

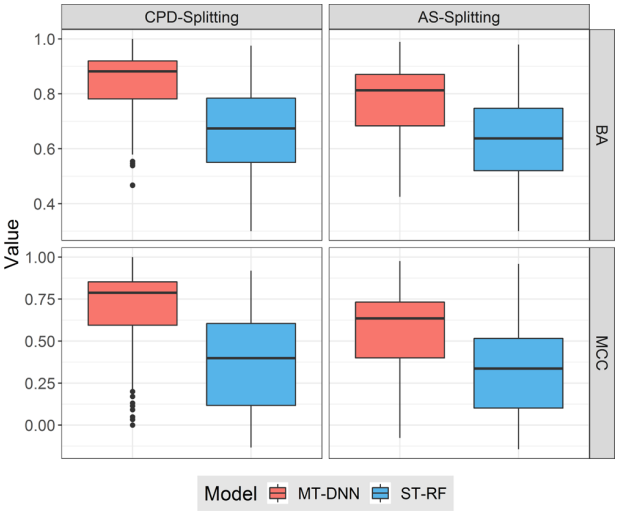


Figure 5. Comparison of MT-DNN and ST-RF models. BA (top) and MCC values (bottom) are reported for MT-DNN (salmon) and ST-RF (blue) predictions using the MACCS fingerprint and test sets obtained by CPD- or AS-splitting.

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

measures and splitting strategies, equivalent trends were observed. With top median BA values exceeding 0.8 and MCC values exceeding 0.75, MT predictions were of high quality.

We note that different from BA, MCC explicitly takes false positives and false negatives into account (see Materials and Methods) and hence provides a balanced quantitative measure of prediction outcomes. Therefore, in the following, ST and MT models were compared on the basis of MCC values.

2.5.2. Comparison of ST- and MT-RF, -SVM, and -DNN Models. A key question was how different ML algorithms would perform under conditions of ST and MT learning. All models were trained and evaluated using data sets obtained by AS-based splitting. In addition to building MT_Algorithm (DNN) and MT_Identity (RF, SVM, DNN) models, as a control, other MT models were also generated following a strategy termed MT_OnlyFP. This approach corresponded to MT_Identity except that kinase identity vectors were omitted. Accordingly, only compound fingerprints were used as features to build a global binary model but no target information such that the predictions completely relied on chemical information.

Figure 6 summarizes our comprehensive comparison of different RF, SVM, and DNN models. For each type of model,

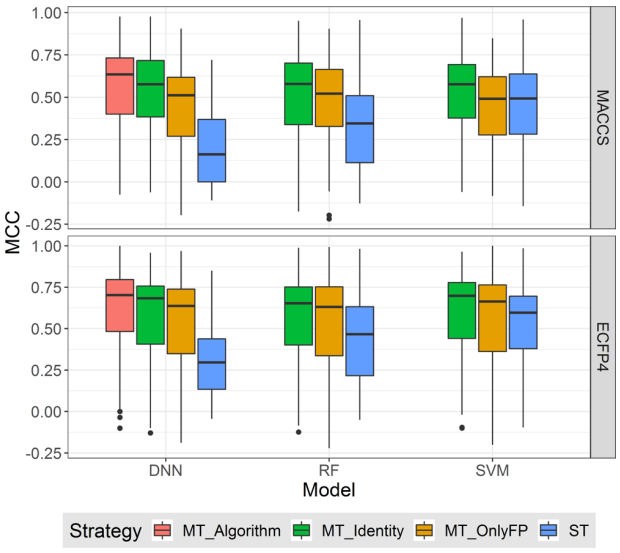


Figure 6. Comparison of MT-ML and ST-ML models. Box plots report distributions of MCC values for per-target predictions for MT- and ST-ML (DNN, RF, and SVM) models using the MACCS (top) and ECFP4 (bottom) fingerprints. Prediction strategies include algorithmic MT-DNN (salmon), MT-ML based on the identity vector (green), MT-ML only with fingerprints (MT_OnlyFP, yellow), and ST-ML (blue).

distributions of MCC values for individual predictions are reported. The performance of ST models was consistently lower than that of MT models. Best ST predictions were obtained using SVM, followed by RF. These models were clearly more accurate than ST-DNN models. Overall best performance was achieved by MT_Algorithm DNN models but only by a very small margin. Surprisingly, the prediction accuracy of MT_Identity models was similar for RF, SVM, and DNN and nearly reached the top performance level. With top median MCC values greater than 0.7, MT predictions achieved overall high accuracy. Moreover, the predictive performance of the MT_OnlyFP models approached the performance level of MT_Identity models for all methods and molecular representations. Thus, chemical features were of critical importance for the predictions, indicating that highly potent inhibitors of a given kinase were often more similar to each other than to inhibitors of other kinases. For RF and DNN, the MT_OnlyFP models were clearly more accurate than the ST models, reflecting partial MT learning capacity, despite the absence of kinase information. For SVM, which produced the best ST models, the performance of MT_OnlyFP models was comparable.

Figure 7 reports true positive (TPR) and false positive rates (FPR) for the different DNN, RF, and SVM modeling

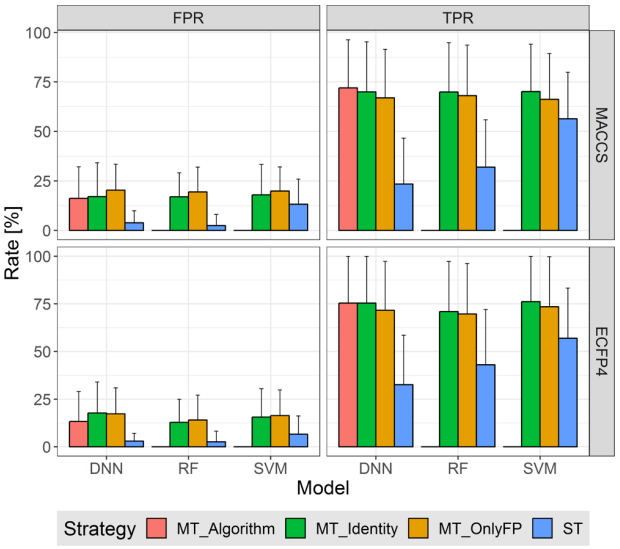


Figure 7. FPR and TPR for MT-ML and ST-ML models, respectively. Mean FPR and TPR plus standard deviation (error bar) are reported for MT- and ST-ML models according to Figure 5.

strategies. The results were comparable to those discussed above. An additional finding was that ST consistently had the lowest FPR and TPR. Thus, compared to MT models, the inferior prediction accuracy of ST models was very likely due to missing highly potent inhibitors, especially for DNN and RF.

Taken together, the results in Figures 6 and 7 revealed that MT learning was largely responsible for achieving high prediction accuracy rather than a specific algorithm. MT-RF, MT-SVM, and MT-DNN models displayed similar performance, although DNN was the only method for which an algorithmic MT implementation was available. On the basis of these observations, the performance of the algorithm-independent MT_Identity strategy was encouraging.

2.6. FW Correlation. In light of the superior performance of MT models for predicting highly potent kinase inhibitors, we further investigated relationships between prediction tasks. Because only 4 and 7% of the highly and weakly potent inhibitors had multikinase annotations, respectively, there was no significant overlap between inhibitor sets for different kinases, which would directly explain correlation effects between prediction tasks. Given this very limited compound overlap, FW correlation was analyzed as an indirect measure of task relatedness. Figure 8a illustrates FW correlation analysis. Because of the sparseness of the compound−target matrix task,

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

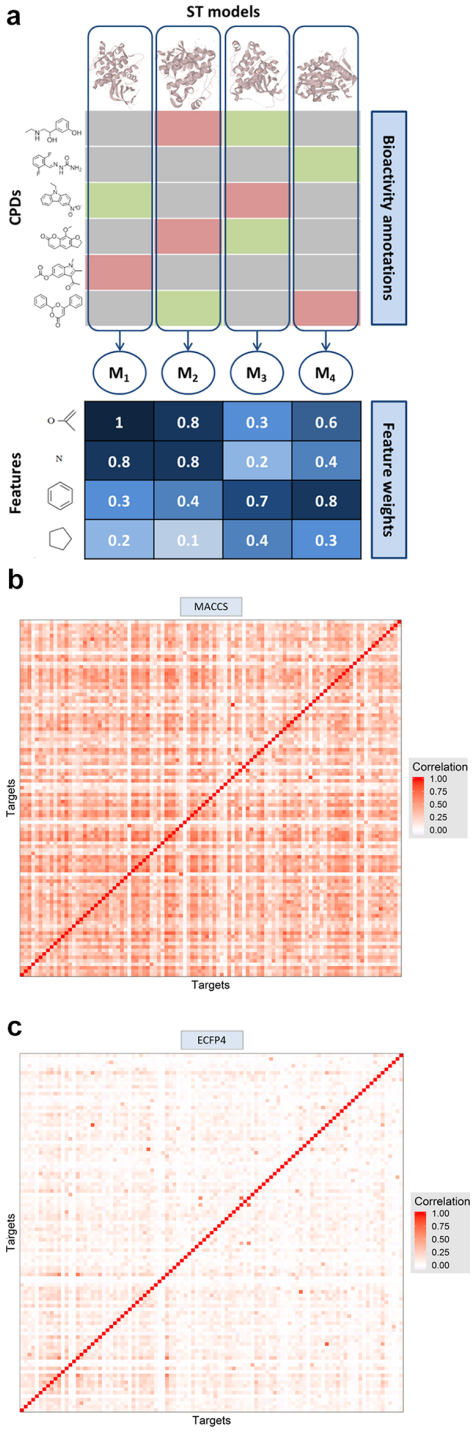


Figure 8. FW correlation for individual tasks. (a) Matrix at the top represents compound−target interactions and label coloring reflects different activity annotations (green: highly potent inhibitor/strong interaction, red: weakly potent inhibitor/weak interaction, gray: missing activity annotation). ST models are generated for individual targets (M_{1-4}). The matrix at the bottom illustrates weighting of different fingerprints features on the basis of the ST models. Pairwise correlation was calculated for FW vectors. The blue color gradient and values from 0.1 to 1 represent the magnitude of FWs. In (b,c), exemplary FW correlation matrices are shown for an individual trial using MACCS and ECFP4, respectively. The color gradient indicates lowest (white) to highest (red) FW correlation.

correlation cannot be determined on the basis of compound activity. As an alternative, ST-RF models were built on a per-target basis to analyze chemical features prioritized for different tasks (targets). FW vectors have the same size for all the targets, corresponding to the dimensionality of the feature space (e.g., 166 features for the MACCS fingerprint), and no missing values. From ST-RF models of all kinases, FWs were extracted on the basis of Gini importance (see Materials and Methods) and pairwise FW correlation among tasks was determined. As a result, a target−target correlation matrix was obtained as shown in Figure 8b,c for an individual trial using MACCS and ECFP4, respectively. Pairwise correlation between FW vectors provided a measure of similarity between prioritized chemical features. Correlation coefficient values significantly varied for different targets (and were overall smaller for ECFP4 than for MACCS).

Then, MT-RF and MT-DNN models were built on the basis of varying numbers of tasks based on FW correlation. Initially, two tasks with the highest (or lowest) FW correlation coefficient were identified and combined. In subsequent iterations, the three tasks with the highest (lowest) mean FW correlation compared to combined tasks were identified and added to an MT model. A total of 10 iterations were carried out such that the number of tasks selected on the basis of maximal or minimal correlation ranged from 2 to 32, yielding two sets of tasks.

Figure 9a (MACCS-based FW correlation) and Figure 9b (ECFP4-based FW correlation) show MCC differences (ΔMCC values) for MT-RF and MT-DNN and corresponding ST models. Value distributions are reported for individual prediction from different iterative trials. Median MCC values were consistently higher for MT models compared to that for ST models and mostly higher for MT models built on the basis of strongly than weakly correlated tasks. Depending on the trial, some tasks might be more correlated than others, resulting in larger performance differences. However, the presence of tasks with strong FW correlation generally supported MT learning and was an at least approximate indicator of MT model performance.

3. CONCLUSIONS

In this study, we have compared ST and MT models for different ML methods including DNN in the prediction of highly potent inhibitors of more than 100 kinases. The data set curated for this analysis provided related prediction tasks. Compared to ST models, MT learning resulted in a performance boost for all methods. Algorithmic MT-DNN models displayed overall highest predictive performance but only by a slight margin compared to MT-RF or MT-SVM models. These models were generated on the basis of the algorithm-independent and generally applicable MT_Identity strategy. This approach combined compound and target information and was effective for MT learning. These findings also indicated that algorithmic transfer learning available for MT-DNN was not required for achieving high predictive performance. Furthermore, control calculations applying the MT_OnlyFP strategy revealed that chemical features of highly potent kinase inhibitors were often sufficient to generate models with at least partial MT character that yielded meaningful predictions. Finally, because only very few shared active compounds were available, FW correlation was determined over a limited number of independent trials and used as a measure of task relatedness. Iteratively selected

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

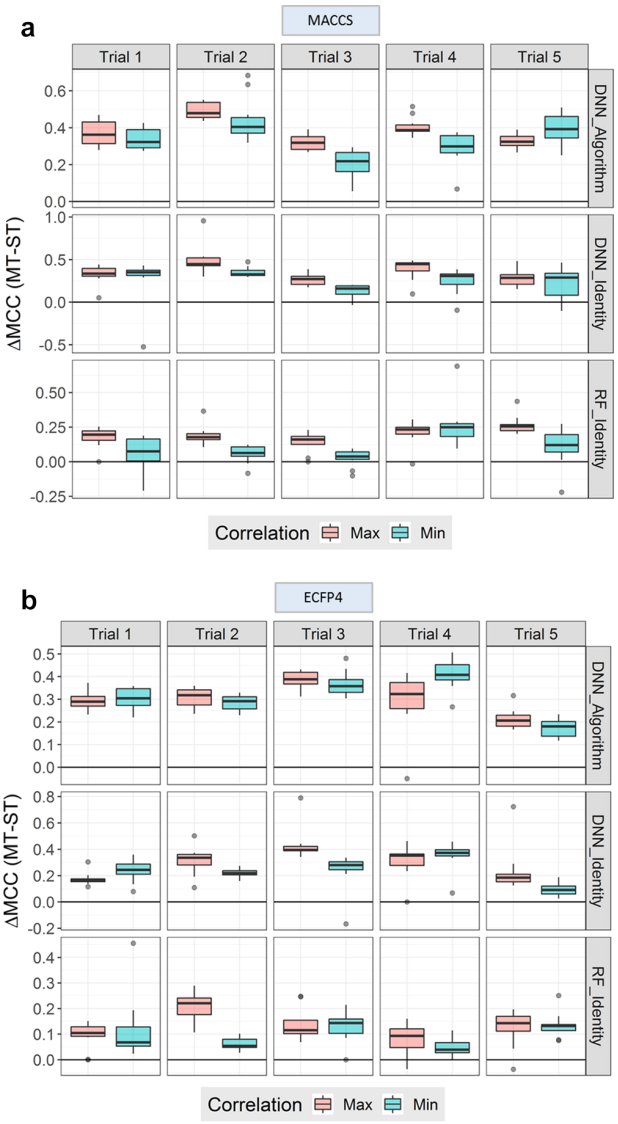


Figure 9. Task addition based on FW correlation. Box plots report distributions of ΔMCC values for comparison of ST-DNN/RF and corresponding MT models. Positive and negative values reflect superior performance of MT and ST models, respectively. MT models with iteratively combined prediction tasks are obtained based on maximal (salmon) or minimal (blue) FW correlation between tasks. ΔMCC values are reported for five trials and three MT models including algorithmic MT-DNN (top) and MT-DNN/RF based on the identity vector (middle/bottom). (a) shows results using MACCS and (b) ECFP4.

prediction tasks prioritized on the basis of FW correlation were found to favor MT learning, which further rationalized observed performance differences between ST- and MT-ML models. Taken together, the findings presented herein assign high priority to MT learning schemes when addressing related prediction tasks. As also shown herein, MT modeling of compound−target interactions does not depend on DNN and can be facilitated with different ML methods. This provides an encouraging perspective for practical applications of MT learning in medicinal chemistry.

4. MATERIALS AND METHODS

4.1. Compound Data Set. *4.1.1. Kinase Inhibitor Data.* Kinase inhibitors with available high-confidence compound activity data were extracted from ChEMBL version 22.[15] A total of 43 331 inhibitors were obtained, which were active against 286 human kinases from 12 different groups, yielding 53 622 pIC_{50} and 5828 pK_i activity annotations.[16] For activity predictions, only pIC_{50} annotations were considered for consistency.

4.1.2. Potency-Based Inhibitor Classification. We distinguished highly potent and weakly potent inhibitors from each other. A potency threshold of $pIC_{50} \geq 8$ (10 nM) was applied to assemble the highly potent (positive) class and a threshold of $pIC_{50} \leq 6$ (1000 nM) to generate the weakly potent (negative) class. Compounds falling into the intermediate potency range were omitted to minimize the influence of boundary effects on predictions. Kinase targets were only selected if at least five positive and five negative compounds were available. On the basis of these criteria, the data set used for predictive modeling comprised a total of 19 030 inhibitors of 103 kinases. These inhibitors formed 11 120 highly potent and 11 252 weakly potent compound−target interactions, yielding an activity annotation density of 1.1% (of all possible kinase−inhibitor interactions). The 19 030 inhibitors contained a subset of 739 compounds with multikinase activity.

4.2. Data Analysis Protocol. *4.2.1. Molecular Representations.* MACCS structural keys[17] and ECFP4[18] were used to represent inhibitors. MACCS consists of a set of 166 predefined structural keys (bits). ECFP4 enumerates layered atom environments and encodes them by integers using a hashing function. The folded version (1024 bits) of ECFP4 obtained by modulo mapping was used. The MACCS and ECFP4 fingerprints were generated using in-house Python scripts based on the *OEChem* toolkit.[19]

4.2.2. Identification of ASs. ASs were systematically identified in the kinase inhibitor set applying computational method[20] based upon the matched molecular pair (MMP) formalism.[21] An MMP is a pair of compounds that only differ by a structural change at a single site. Here, MMPs were generated by bond fragmentation on the basis of retrosynthetic combinatorial analysis procedure rules.[22] A total of 2117 ASs with two or more inhibitors were identified. These series contained a total of 12 290 compounds. The remaining inhibitors were singletons that were treated as individual entities (compounds or series) for the generation of training and test sets.

4.2.3. Model Building and Evaluation. Training and test sets were created for model building and evaluation, respectively.[23,24] Two strategies were applied to separate compounds into training and test sets including compound-based splitting and AS-based splitting, as illustrated in Figure 2. Compound-based splitting assigned individual compounds to the training or test set, whereas the AS-based approach separated complete ASs. Thus, in the latter case, models were trained and tested on different ASs. To avoid potential AS annotation bias,[7,25] AS-based splitting assigned all kinase annotations of analogs either to the training or to the test set. The proportion of compounds or ASs in training and test sets was constantly set to 75% versus 25%. For each splitting strategy, five independent trials were carried out.

4.2.4. Parameter Optimization and Performance Measures. For each ML method, trial, and splitting strategy, model

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

hyperparameters were optimized through a twofold cross-validation on the training set. Following optimization, the final model was built using the complete training set.

As performance measures, BA[26] and MCC[27] values were calculated for test set predictions

$$\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

4.3. ML Algorithms. Three different classification algorithms were used for predictive modeling including RF, SVM, and DNN.

4.3.1. Random Forest. RF represents an ensemble of decision trees trained on a bootstrap sample of the data set. At each node, the best data split is selected for a random feature subset, known as feature bagging.[28] The final prediction is a consensus over all decision trees. The ensemble approach reduces the risk of overfitting of individual trees. Moreover, feature bagging avoids a high correlation between trees. RF models were built using 500 trees and optimizing three hyperparameters: (i) the minimum number of samples per leaf node (with candidate values of 1, 5, and 10); (ii) the maximum number of features at each node (all features, the square root, or the logarithm to the base 2 of the total number of features); and (iii) presence or absence of class weights. The application of class weights permits preferential penalization of errors in the minority class during training. Table 1 reports most frequently selected hyperparameters during optimization. The *scikit-learn* implementation was used[14] (with remaining hyperparameters set to default values).

Table 1. Optimal Hyperparameters for RF[a]

strategy	fingerprint	min. leaf samples	max. features	class weights
MT_Identity	MACCS	1	square root, log 2	yes
MT_Identity	ECFP4	1	square root, log 2	yes, no
ST	MACCS	1	square root	no
ST	ECFP4	1	square root	no

[a]Reported are the most frequently selected optimal hyperparameters for RF when different modeling strategies and fingerprint descriptors were used. MT_Identity refers to the descriptor-based strategy for MT learning (see Section 4.4.2).

4.3.2. Support Vector Machines. SVM is a kernel-based classification algorithm that defines a hyperplane H capable of separating two classes in the input space $\mathcal{X}$.[29] The hyperplane is defined by the expression $H = \{\mathbf{x}|\langle \mathbf{w},\mathbf{x}\rangle + b = 0\}$, where $\mathbf{w}$ is the weight vector and b a bias obtained by maximizing the margin (or distance) between the training classes. Because a hard-margin model is prone to overfitting, a certain number of misclassifications can be allowed by adding slack variables. The cost or regularization hyperparameter C controls the amount of instances falling within the margin or on the incorrect side of the hyperplane (i.e., the training errors). Candidate C values during internal cross-validation were 0.001, 0.01, 0.1, 1, 10, 100, and 1000. Hyperparameter optimization is summarized in Table 2.

Table 2. Optimal Hyperparameters for SVM[a]

strategy	fingerprint	cost	class weights
MT_Identity	MACCS	100	yes, no
MT_Identity	ECFP4	100	yes, no
ST	MACCS	10	yes
ST	ECFP4	10	yes

[a]Reported are the most frequently selected optimal hyperparameters for SVM when different modeling strategies and fingerprint descriptors were used. MT_Identity refers to the descriptor-based strategy for MT learning (see Section 4.4.2).

If linear separation of data with different class labels is not possible in a given feature space, nonlinear kernel functions are applied to map the data into a higher-dimensional space where linear separation might be possible. Through this "kernel trick" computing, an explicit mapping can be avoided.[30] SVM calculations were carried out with *scikit-learn* using the Tanimoto kernel.[31,32]

4.3.3. Deep Neural Networks. Feed-forward DNNs consist of a collection of connected units or neurons that are organized in different layers such that the output of one layer is used as the input of the next. In addition to an input and output layer, DNNs have at least two hidden layers.[33] Thus, a DNN model can be understood as a series of functional transformations.[34] Each hidden neuron applies a nonlinear activation function to the weighted sum of its inputs,[35] according to the following equation:

$$y_j = f\left(\sum_{i=1}^{d} x_i \omega_{ji} + \omega_{j0}\right)$$

Here, f is the activation function; i and j account for unit indices on the input and hidden layers, respectively; ω_{ji} indicates the weights at the hidden unit j; ω_{j0} represents the biases; x_i is the input; and y_j is the output of neuron j.

During training, the weights of the network are adjusted such that the input matches the desired outputs and the loss is minimized. The gradient of the loss function was computed using the backpropagation algorithm,[33] and weights were varied subsequently to minimize the loss (gradient descent). DNNs were built with a single unit or multiple units in the output layer, yielding ST- and MT-DNN models, respectively. DNNs were calculated using *Tensorflow*[36] and *Keras*.[37]

DNNs contain more hyperparameters than RF and SVM, and predictions are generally more dependent on prior optimization. The optimum architecture (i.e., number of hidden layers and neurons per layer) was chosen from the following options: [200,100], [2000,1000], [200,100,100] for ST-DNN and [200,100], [2000,1000], [2000,1000,100] for MT-DNN. Candidate values for the learning rate were 0.01 and 0.001 for both ST- and MT-DNN, respectively, and models with and without class weights were considered. For ST-DNN, batch sizes were 128 or 256, and "Adam" was used as the optimization algorithm. For MT-DNN, the batch size was set to 258, and the optimizer was either Adam or stochastic gradient descent with momentum. The activation function was the rectified linear unit, except for the output layers for which softmax (ST-DNN) or sigmoid (MT-DNN) functions were used. A dropout rate of 25% was permitted to prevent overfitting. For internal validation, 100 epochs were applied; for external validation, the best model was retained

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

Table 3. Optimal Hyperparameters for DNN[a]

strategy	fingerprint	architecture [hidden layers]	learning rate	batch size	class weights	optimizer
MT_Algorithm	MACCS	[2000,1000,100]	0.001	(256)	no	Adam
MT_Algorithm	ECFP4	[2000,1000]	0.001	(256)	yes	Adam
MT_Identity	MACCS	[2000,1000]	0.001	(256)	yes	Adam
MT_Identity	ECFP4	[2000,1000,100]	0.001	(256)	no	Adam
ST	MACCS	[200,100,100],[2000,1000]	0.01, 0.001	128, 256	no	Adam
ST	ECFP4	[200,100,100]	0.01, 0.001	128, 256	no	Adam

[a]Reported are the most frequently selected optimal hyperparameters for DNN when different modeling strategies and fingerprint descriptors were used. Hyperparameters in parentheses were not optimized. MT_Algorithm refers to the algorithm-based strategy for MT learning (see Section 4.4.1) and MT.Identity to the descriptor-based strategy (see Section 4.4.2).

after 500 epochs. Table 3 summarizes the most frequently selected hyperparameters for different DNN models.

4.4. MT Implementations. For MT learning, two strategies were considered. First, algorithmic adaptations were carried out such that MT learning was embedded into the algorithm.[9] Second, an algorithm-independent approach was applied based on descriptor modification.

4.4.1. Algorithm-Based Approach. Classification algorithms might be modified to support MT modeling. For RF and SVM, no MT implementations are currently available, which can handle missing labels. For algorithm-based MT-DNNs, 103 output units were used (one per kinase). A customized loss function was created to mask missing labels and only use labeled examples when computing cross-entropy loss.

4.4.2. Descriptor-Based Approach. The MT_Identity strategy was applied to all methods. An identity vector was generated to encode the 103 targets (one bit per target) potentially involved in inhibitor−kinase interactions and concatenated with the compound fingerprint. The resulting MACCS- and ECFP4-based hybrid fingerprints then consisted of 269 (166 + 103) and 1127 (1024 + 103) bits, respectively. In the identity vector, only one bit was set on. Then, a binary model was built to distinguish between highly potent and weakly potent compound−kinase interactions.

4.5. Feature Weighting. ST-RF models were used to obtain weights for each feature on the basis of the Gini importance.[38] The Gini impurity index estimates the quality of a data split at a given node. The lower the Gini index value, the larger the influence of a feature on a data split. FWs were then obtained by summing over all nodes including a given feature proportionally to the number of examples.

AUTHOR INFORMATION

Corresponding Author

*E-mail: bajorath@bit.uni-bonn.de. Phone: 49-228-7369-100 (J.B.).

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Author Contributions

This study was carried out and the manuscript was written with contributions of all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 676434, "Big Data in Chemistry" ("BIG-CHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) is responsible for any use that may be made of the information it contains. The authors thank OpenEye Scientific Software, Inc., for providing a free academic license for the OpenEye toolkit and also thank Nils Weskamp for support and helpful discussions.

REFERENCES

(1) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Chemoinformatics: Quo Vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413−1437.

(2) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3−14.

(3) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.

(4) Baskin, I. I.; Winkler, D.; Tetko, I. V. A Renaissance of Neural Networks in Drug Discovery. *Expert Opin. Drug Discovery* **2016**, *11*, 785−795.

(5) Ekins, S. The Next Era: Deep Learning in Pharmaceutical Research. *Pharm. Res.* **2016**, *33*, 2594−2603.

(6) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.

(7) 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*, 5441−5451.

(8) Caruana, R. Multitask Learning. In *Learning to Learn*; Thrun, S., Pratt, L., Eds.; Springer: New York, 1998; pp 95−133.

(9) Zhang, M.; Zhou, Z. A Review on Multi-Label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1819−1837.

(10) Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier Chains for Multi-label Classification. *Mach. Learn.* **2011**, *85*, 333−359.

(11) Xu, Y.; Ma, J.; Liaw, A.; Sheridan, R. P.; Svetnik, V. Demystifying Multitask Deep Neural Networks for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model.* **2017**, *57*, 2490−2504.

(12) Rodríguez-Pérez, R.; Miyao, T.; Jasial, S.; Vogt, M.; Bajorath, J. Prediction of Compound Profiling Matrices Using Machine Learning. *ACS Omega* **2018**, *3*, 4713−4723.

(13) Rodríguez-Pérez, R.; Bajorath, J. Prediction of Compound Profiling Matrices, Part II: Relative Performance of Multitask Deep Learning and Random Forest Classification on the Basis of Varying Amounts of Training Data. *ACS Omega* **2018**, *3*, 12033−12040.

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

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

(15) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100.
(16) Dimova, D.; Bajorath, J. Assessing Scaffold Diversity of Kinase Inhibitors Using Alternative Scaffold Concepts and Estimating the Scaffold Hopping Potential for Different Kinases. *Molecules* **2017**, *22*, 730−740.
(17) *MACCS Structural Keys*; Accelrys: San Diego, CA, 2011.
(18) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(19) *OEChem TK*, version 2.0.0; OpenEye Scientific Software: Santa Fe, NM, 2015.
(20) Stumpfe, D.; Dimova, D.; Bajorath, J. Computational Method for the Systematic Identification of Analog Series and Key Compounds Representing Series and Their Biological Activity Profiles. *J. Med. Chem.* **2016**, *59*, 7667−7676.
(21) Kenny, P. W.; Sadowski, J. Structure Modification in Chemical Databases. In *Cheminformatics in Drug Discovery*; Oprea, T. I., Ed.; Wiley-VCH: Weinheim, Germany, 2004; pp 271−285.
(22) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAPRetrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511−522.
(23) Stone, M. Cross-validatory Choice and Assessment of Statistical Predictions. *J. Roy. Stat. Soc. B Methodol.* **1974**, *36*, 111−133.
(24) Baumann, D.; Baumann, K. Reliable Estimation of Prediction Errors for QSAR Models under Model Uncertainty Using Double Cross-validation. *J. Cheminf.* **2014**, *6*, No. e47.
(25) Scior, T.; Bender, A.; Tresadern, G.; Medina-Franco, J. L.; Martínez-Mayorga, K.; Langer, T.; Cuanalo-Contreras, K.; Agrafiotis, D. K. Recognizing Pitfalls in Virtual Screening: A Critical Review. *J. Chem. Inf. Model.* **2012**, *52*, 867−881.
(26) Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; Buhmann, J. M. The Balanced Accuracy and Its Posterior Distribution. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, 2010; pp 3121−3124.
(27) Matthews, B. W. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta, Protein Struct.* **1975**, *405*, 442−451.
(28) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(29) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.
(30) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, Pittsburgh, PA, 1992; pp 144−152.
(31) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Networks* **2005**, *18*, 1093−1110.
(32) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-based Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.
(33) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015.
(34) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.
(35) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*, 2nd ed.; John Wiley & Sons: New York, 2000.
(36) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; Zheng, X. *TensorFlow: A System for Large-scale Machine Learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, 2016.
(37) Chollet, F. Keras, version 2.1.3, 2015. https://github.com/keras-team/keras (accessed Jan 17, 2018).
(38) Hastie, T.; Tibshirani, R.; Friedman, J. H. *The Elements of Statistical Learning*; Springer: Berlin, 2009.

DOI: 10.1021/acsomega.9b00298
ACS Omega 2019, 4, 4367−4375

Summary

Strong and weak inhibitors against multiple kinases were successfully differentiated by MT learning. In particular, MT-DNNs provided the best results indicating that MT learning might be favored due to the inherent relation among the tasks. However, MT-DNN model performance was closely followed by a very simplistic method that allowed standard ML models to use all ligand data to train a single model. In this approach, an interaction is codified by the concatenation of fingerprint (inhibitor) and a one-hot encoding (kinase), and a model is trained to classify strong and weak interactions. Overall, MT-DNN did not extract much additional information compared to this simplistic approach (MT-Identity) implemented with other ML algorithms. Even though additional protein information was not available, MT-Identity models also provided significant superior performance compared to ST. This suggests that prioritized chemical patterns were mostly common to distinct kinases. Thus, the calculation of task correlation based on model feature weights was proposed for non-overlapping activity measurements across targets.

The next two chapters include investigations about relevant methodological aspects in bioactivity predictions, namely the influence of the nature of training compound data on ML model performance.

Chapter 4

Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds

Introduction

The previous chapters have presented two successful applications of ML algorithms for the prediction of ligand-protein interactions against multiple biological targets including data from different sources and thus distinct nature. In supervised learning, training data essentially determines the quality and predictive ability of a model. Herein, the effect of training set size in activity predictions is systematically studied. Furthermore, activity data from medicinal chemistry sources are characterized by large sets of active compounds and only few inactive annotations, whereas screening data shows the opposite trend and often contains only small numbers of valid hits. Consequently, training set composition is also investigated. Calculations are carried out using the state-of-art SVM method, which is widely used in pharmaceutical research and allows both linear and non-linear modeling. SVM-based compound activity predictions are analyzed for distinct activity classes from ChEMBL.

JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

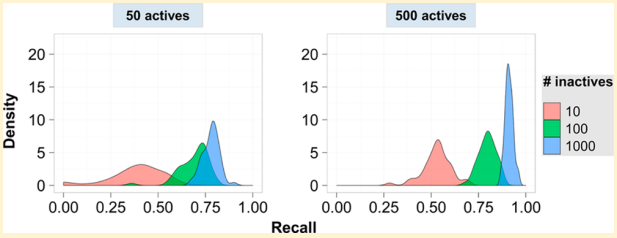
pubs.acs.org/jcim

Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds

Raquel Rodríguez-Pérez, Martin Vogt, and Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstrasse 2, D-53113 Bonn, Germany

ABSTRACT: Support vector machine (SVM) modeling is one of the most popular machine learning approaches in chemoinformatics and drug design. The influence of training set composition and size on predictions currently is an underinvestigated issue in SVM modeling. In this study, we have derived SVM classification and ranking models for a variety of compound activity classes under systematic variation of the number of positive and negative training examples. With increasing numbers of negative training compounds, SVM classification calculations became increasingly accurate and stable. However, this was only the case if a required threshold of positive training examples was also reached. In addition, consideration of class weights and optimization of cost factors substantially aided in balancing the calculations for increasing numbers of negative training examples. Taken together, the results of our analysis have practical implications for SVM learning and the prediction of active compounds. For all compound classes under study, top recall performance and independence of compound recall of training set composition was achieved when 250−500 active and 500−1000 randomly selected inactive training instances were used. However, as long as ∼50 known active compounds were available for training, increasing numbers of 500−1000 randomly selected negative training examples significantly improved model performance and gave very similar results for different training sets.



INTRODUCTION

The support vector machine (SVM) algorithm[1,2] is among the most widely used supervised machine learning methods in chemoinformatics and computer-aided drug discovery.[3−5] The popularity of SVM modeling primarily stems from generally high predictive performance in compound classification and virtual screening.[4] Although SVMs have been applied to investigate a variety of class label prediction and also regression tasks in chemoinformatics and drug discovery research,[4,5] so far only very few studies have addressed the issue of training set composition and size for SVM modeling[6] and other machine learning methods.[7,8] Especially the choice of negative training examples is often little considered in machine learning. Typically, to train models for compound classification, a subjectively chosen number of molecules are randomly selected from chemical databases to serve as negative training instances, without further analysis. Two previous studies have investigated the choice of negative training examples in greater detail.[6,7] For SVM modeling, the use of experimentally confirmed negative training compounds from screening assays and randomly chosen compounds from the ZINC database[9] was compared in the prediction of active compounds.[6] It was shown that the source of negative training instances affected the performance of SVM classification. Perhaps surprisingly, randomly selected ZINC compounds often resulted in better models than screening compounds that were confirmed to be inactive against a target for which active compounds were predicted.[6] No training set variations were carried out. In another study, negative training sets were assembled from different databases for compound classification using different machine learning approaches.[7] These calculations revealed a notable influence of negative training examples on the predictions and a preference for randomly selected ZINC compounds over compounds from other sources.[7] In this case, the size of negative training sets was varied when building models using different machine learning methods including SVMs with polynomial kernels. Training set size variations were found to influence compound predictions.[7] Performance relationships for varying numbers of negative and positive training examples were not investigated. In other studies, positive and negative training examples were balanced to improve the performance of machine learning models,[6,8] addressing the issue of data imbalance in machine learning.[10,11]

Herein, we report an analysis of the influence of training set composition and size on SVM classification and ranking by systematically varying the number of negative and positive training examples and determining how these variations affect the prediction of active compounds and stability of the calculations.

MATERIALS AND METHODS

SVM Classification. For SVM classification,[1] training compounds are defined by a feature vector $\mathbf{x} \in \mathcal{X}$ and a class label $\gamma \in \{-1, 1\}$ and projected into the reference space $\mathcal{X}$. SVMs solve a convex quadratic optimization problem to find a

Received: February 15, 2017
Published: April 4, 2017

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

hyperplane $H = \{\mathbf{x}|\langle \mathbf{w}, \mathbf{x}\rangle + b = 0\}$ that separates the positive and negative class. The hyperplane H is defined by a normal vector $\mathbf{w}$ and a bias b and maximizes the margin between the two classes. To achieve model generalization, non-negative slack variables ξ_i are considered during training to penalize misclassification. In addition, the cost hyperparameter C controls the trade-off between margin maximization and permitted training errors, and its value can be optimized by cross-validation.[12]

Once the decision boundary is defined, test instances are projected into the feature space. New compounds of unknown class label are classified according to the side of the hyperplane on which they fall or, alternatively, ranked according to the value of $g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle$.[13] The latter strategy is equivalent to changing the bias of the hyperplane, sliding it from the most distant points on the positive side toward the negative side, and ranking compounds in the order they pass through the plane.

In the case of nonlinearly separable training data in a given reference space, the scalar product <·, ·> can be replaced by a kernel function K(·, ·), which is known as the *kernel trick*.[14] Using kernel functions, the scalar product of two feature vectors can be computed in a higher dimensional space $\mathcal{H}$ where the data may be linearly separable without the need to explicitly compute the mapping of $\mathcal{X}$ into $\mathcal{H}$. In SVM-based compound classification, the Tanimoto kernel is one of the most frequently used kernel functions for binary fingerprints.[15]

For imbalanced data sets, different class weights can be assigned to put relative weights on misclassification of positive and negative training instances and avoid orienting the hyperplane toward the minority class. Accordingly, C_+ and C_- balance the weight on slack variables for the positive and negative class, respectively.[16]

$$\frac{C_+}{C_-} = \frac{|\{i|y_i = -1\}|}{|\{i|y_i = +1\}|}$$

Compound Data Sets and Representation. Ten sets with at least 600 active compounds (positive instances) were obtained from ChEMBL version 22.[17] Only compounds with numerically specified equilibrium constants (K_i values) for single human proteins were selected, while omitting borderline active compounds ($pK_i < 5$) that might often represent artifacts. Table 1 reports the accession number, target name, number of compounds and mean pK_i values for these 10 compound data sets. As background set (pool of negative instances), 250 000 compounds were randomly selected from ZINC.[9] Random subsets of these compounds were used as negative training and test examples. For model building, all active and inactive compounds were represented as standard MACCS fingerprints[18] consisting of 166 bits monitoring the presence (bit set on) or absence (set off) of predefined structural fragments or patterns. Although we deliberately selected the simplistic and easy to rationalize MACCS fingerprint for our proof-of-concept investigation, control calculations were also carried out using the folded version of the extended connectivity fingerprint with bond diameter 4 (ECFP4).[19]

Table 1. Compound Data Sets[a]

accession no.	target name	number of compounds	mean pK_i
P00734	thrombin	839	6.67
P00918	carbonic anhydrase 2	2164	7.22
P21917	dopamine D4 receptor	804	7.11
P41146	nociceptin receptor	844	7.81
P00742	coagulation factor X	1476	7.77
P29275	adenosine receptor A2b	1187	7.12
P32245	melanocortin receptor 4	1260	7.00
Q9H3N8	histamine H4 receptor	875	6.97
Q99705	melanin-concentrating hormone receptor 1	1208	7.45
Q9Y5Y4	prostaglandin D2 receptor 2	833	7.53

[a]Ten compound data sets were selected from ChEMBL and used for SVM modeling. For each activity class, the ChEMBL accession no., target name, number of compounds, and mean pK_i value are reported.

Calculation Protocol.

(1) Each activity class was randomly divided into training and test (prediction) sets. Training set size was varied across values #I = {10, 50, 100, 500, 1000} for the negative (inactive) class and #A = {10, 50, 100, 250, 500} for the positive (active) class. Test sets always consisted of 10 000 inactive and 100 active compounds.

(2) Preprocessing of the fingerprints of the training and test data was carried out by removing zero-variance features and applying centering and unit variance scaling to all features on the basis of the training set for each trial.

(3) For each of the 25 training set combinations, SVM models were built using the linear and Tanimoto kernel with class weights C_+ and C_-. In addition, cost factors C controlling the influence of individual support vectors were optimized using values of 0.01, 0.1, 1, and 10. For cost factor optimization, 10-fold cross-validation was carried out with training data splits of 60% (model derivation) and 40% (testing, internal validation). Models with best cost factors were selected on the basis of largest area under the ROC curve (AUC).

(4) The optimized SVM model was used to rank test set compounds in the order of decreasing probability of activity based upon the signed distance from the hyperplane (positive to negative side). Model performance was assessed by determining the recall rate of active compounds within the top 1% of ranked test compounds. In addition, balanced accuracy (BA) was calculated, defined as

$$\mathrm{BA} = \frac{0.5\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{0.5\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

(TP, true positives; TN, true negatives; FP, false positives; FN, false negatives).

(5) For each activity class and combination of a kernel function and training set size, the modeling process was carried out 50 times to obtain a distribution of recall rates.

(6) The results were compared using hypothesis testing. The nonparametric Kolmogorov–Smirnov test[20] was employed to account for differences between cumulative recall distributions and the Levene test[21] to compare the variance of these distributions. In addition, the Bonferroni correction[22] was introduced for multiple testing.

The calculation protocol was implemented in R,[23] and the *kernlab* package[24] was used for SVM modeling.

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

RESULTS AND DISCUSSION

For different activity classes, SVM classification and ranking models were built under systematic variation of training set composition and size and active compounds were predicted. Specifically, the number of negative and positive training examples was varied in the ranges of 10−1000 and 10−500, respectively, and all possible combinations were explored. In addition, cost factors were optimized by cross-validation and class-specific weights were used to account for data imbalance in the training set.

Class Weights. Figure 1 compares balanced accuracy of the predictions in the presence or absence of class weights for two representative activity classes. Consideration of class-specific weights consistently improved the accuracy of the predictions for imbalanced training sets, except for three cases of large training sets with at least 250 actives and 500 inactives for which the performance was comparable. Hence, the explicit consideration of different class weights for positive and negative training instances produced more accurate classification models. Under these conditions, the derived hyperplane was not skewed toward the minority class, resulting in improved model generalization, especially in the presence of large training data imbalance. These effects were outweighed only for the largest and least imbalanced training sets. Given the demonstrated relevance of class weights for prediction accuracy, a factor that is not always considered in SVM modeling, results reported in the following included class weight settings.

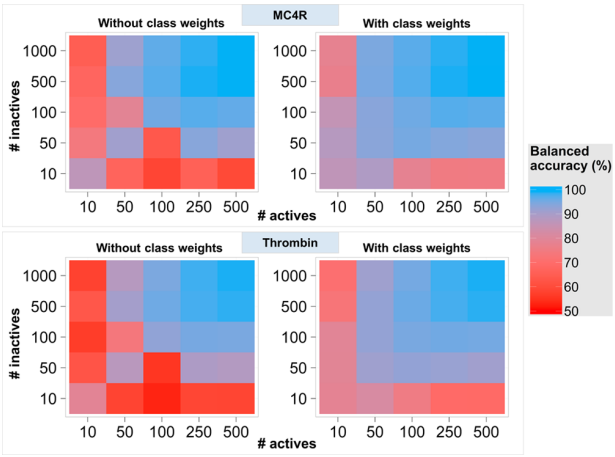


Figure 1. Effects of class weights on model performance. Heat map representations show balanced accuracy over 50 independent trials (using a two-color gradient) for training sets of varying composition and size: (top) melanocortin receptor 4 (MC4R) ligands, (bottom) thrombin inhibitors.

In addition, optimization of cost factors was carried out using cross validation. The best cost factors often varied depending on training set composition, but for well-performing training sets (i.e., those with large numbers of actives and inactives), there was an overall preference for C values of 0.01 for both the linear and Tanimoto kernels. For highly imbalanced data sets, larger cost factors were frequently selected, indicating that adjusting margin softness (stability) also contributed to model generalization. It is noteworthy that for different training set compositions and regardless of the cost factor chosen the hyperplanes generated by the SVMs were very frequently able to separate the training data without error and thus resulted in a hard margin classifier.

Kernels and Fingerprints. Figure 2 reports compound recall for alternative kernel functions under systematic variation of inactive and active training instances for two representative activity classes. Figure 3 shows corresponding density plots for recall rate distributions over multiple trials. First, we focus on relative kernel performance. The results in Figure 2 and 3 reveal generally higher recall performance for the Tanimoto than the linear kernel, frequently reaching a recall level of 0.9. However, even for the linear kernel, satisfactory recall was observed, often approaching a recall level of 0.75. Differences in recall performance between the linear and Tanimoto kernel were quantitatively assessed for all activity classes and statistically compared using the two-sided and paired Kolmogorov−Smirnov test. The results confirmed that the Tanimoto kernel generally performed significantly better than the linear kernel for training instances of #A = {100, 250, 500} and #I = {100, 500, 1000}. However, there was no significant difference in the cases of #A = {10} and #I = {50, 100, 500, 1000} where prediction accuracy was limited. Furthermore, as shown in Figure 3, SVM models derived using the Tanimoto kernel were generally more robust, i.e., corresponding recall rate distributions were sharper for the Tanimoto than for linear kernel. The presence of narrow distributions indicated that models derived from different training sets had comparable prediction accuracy for alternative test instances. As a control, SVM calculations were also repeated using the radial basis function (RBF) kernel,[25,26] another popular kernel function, with a sigma setting, corresponding to the inverse kernel width, of 0.01.[26] The results obtained using the RBF kernel were, on average, nearly indistinguishable from those obtained using the Tanimoto kernel discussed in the following. As an additional control, the calculations were also carried out using ECFP4 instead of MACCS to compare the trends observed for training set variation. With both fingerprints, the same trends were observed (with the typical slightly better recall performance of ECFP4 relative to MACCS).

Training Sets of Varying Composition and Size. The results in Figures 2, 3, and 4 revealed two key findings; (i) recall performance and model generalization consistently improved with increasing size of training sets and (ii) the ratio of active vs inactive training examples significantly influenced prediction accuracy. The increases in recall performance observed in Figure 2 were detected for all activity classes. When the number of active training instances was kept constant, recall rates increased with increasing numbers of inactive instances, except in the case of 10 actives, where prediction accuracy was generally low even over the range of 100−1000 negative instances. Thus, a minimum number of active training compounds was required for training sets of increasing size. Similar observations were made when the number of inactive training compounds was kept constant and the number of active examples was increased. Ten negative examples were consistently insufficient for building effective models and 50 negative training instances were often insufficient (Figure 2). However, in the presence of at least 100 negative training instances, high prediction accuracy was consistently achieved when the number of active examples was increased (Figure 3).

For all compound classes, incremental increase in the number of negative (positive) training instances led to systematic performance enhancements when at least 50 positive (100 negative) training compounds were used, as confirmed by

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

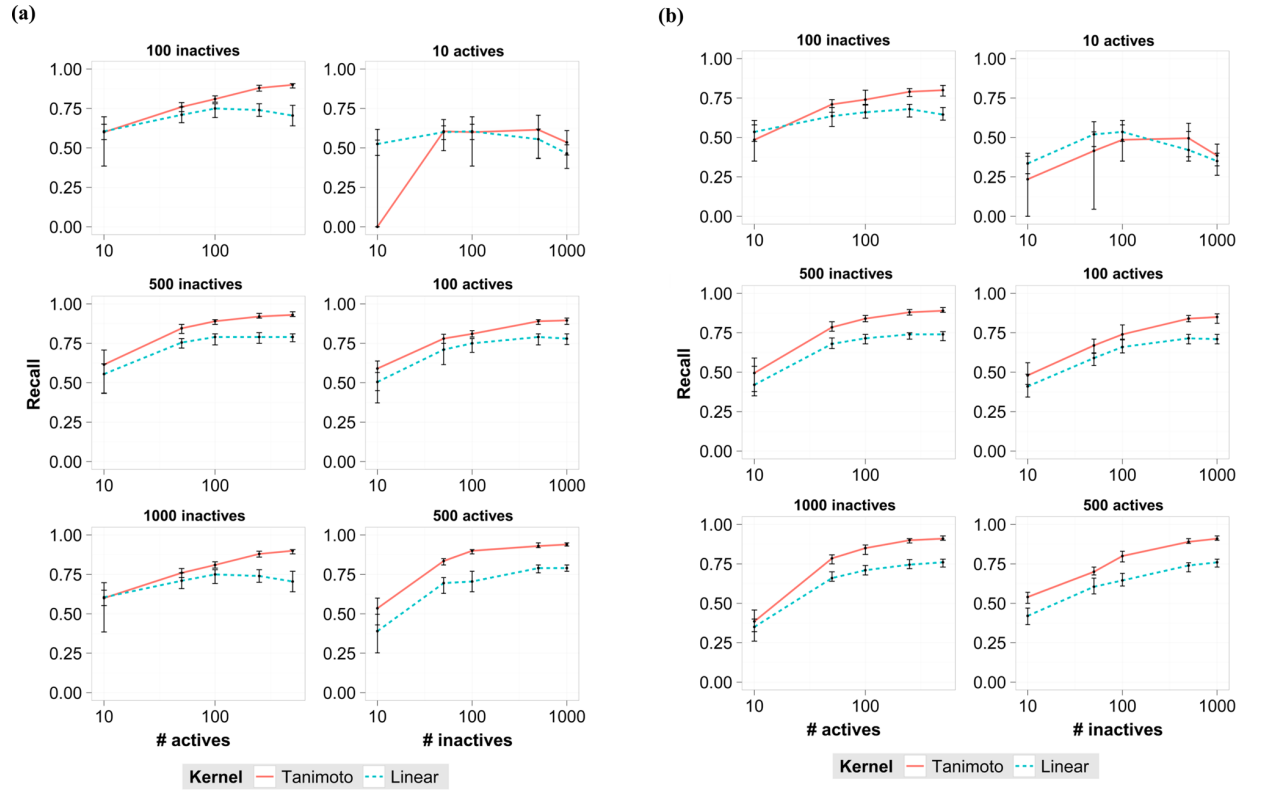


Figure 2. Recall performance. The median value and interquartile range of the recall rate of active compounds among the top 1% of the ranking is reported for 50 trials with the linear (blue dashed line) or Tanimoto (red solid line) kernel. Results monitor the evolution of recall for a constant number of inactives (or actives) and increasing number of actives (or inactives) in the training set: (a) melanocortin receptor 4 ligands, (b) thrombin inhibitors.

the one-sided Kolmogorov−Smirnov test. While overall highest prediction accuracy was achieved for training sets consisting of 500 active and 1000 inactive examples, similar accuracy was already observed for 100 active and 500 inactive training compounds. Furthermore, recall generally began to reach a plateau when at least 100 active and 500 inactive training instances were used (Figure 2). However, with further increasing training set size, recall rate distributions became narrower, as illustrated in Figure 3 and 4, which was indicative of models with consistent prediction accuracy despite training set variations, as mentioned above.

Table 2 compares the recall performance over all activity classes for one of the worst and the best performing training set compositions of 10 actives/100 inactives and 500 actives/1000 inactives, respectively. In the bad case scenario, recall rates of compounds were—with one exception—lower than 50% with large standard deviations and balanced accuracy was around the 80% level. By contrast, for the best performing large training sets, recall rates were consistently high, with a mean of 87%, and balanced accuracy was approaching 100% with very low standard deviations (Table 2). Interestingly, training set imbalance only limited the accuracy of predictions in the case of small but not large training sets, as illustrated in Figure 4, an effect that can be ascribed to the use of class weights for SVM models, as detailed above. For example, while an inactive vs active ratio of 10:1 produced inaccurate predictions for training sets comprising 100 inactive and 10 active training examples, prediction accuracy was high when 1000 inactive and 100 active training compounds were used. Similar observations were made for other compound ratios.

Variance. Taken together, the results in Figure 3 and 4 clearly indicate that the predictions became stable with increasing size of training sets, another key finding. Figure 5 reports the variance of recall rates over independent predictions using training sets of increasing size and provides confirmatory evidence. Furthermore, Levene tests for all activity classes confirmed that the variance of recall distributions significantly differed in 38 of 40 cases (resulting from 10 compound classes and four training set conditions) when training sets with at least 50 active and 10 or 1000 inactive examples were used. By contrast, no statistically significant differences in variance of recall rate distributions were detected when the SVM models were trained with 100 or 1000 inactive examples, regardless of the number of actives.

■ CONCLUSIONS

Herein, we have systematically analyzed the influence of training set composition and size on the prediction accuracy of SVM classification models. Different from earlier studies, our calculations have stressed the importance of considering class weights and optimizing cost factors when imbalanced training sets are used. Furthermore, the ratio of active vs inactive training examples substantially affected the ability of SVM models to correctly predict active compounds. However, recall rates and balanced accuracy consistently improved for training sets of increasing size for all compound classes under study. Increasing size of training sets also compensated for inherent

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

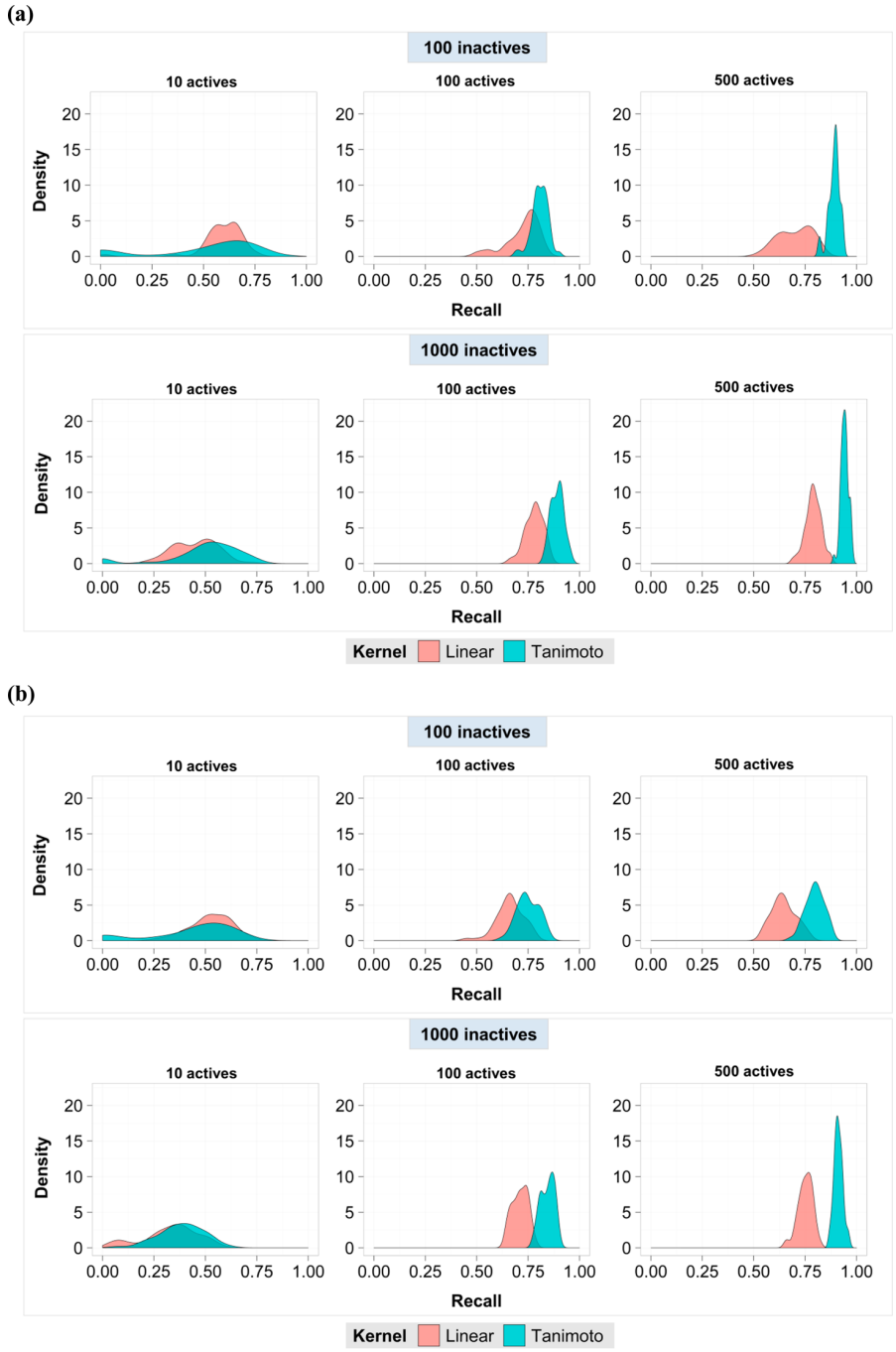


Figure 3. Density estimates. The distribution of recall rates over 50 trials is given for 100 (top) and 1000 (bottom) inactive and increasing numbers of active training compounds: (a) melanocortin receptor 4 ligands, (b) thrombin inhibitors.

data imbalance. Moreover, large training sets led to robust predictions and the accuracy was essentially constant when different training sets of the same size were used. Taken together, our findings have implications for practical applications of SVM classifiers. The following conclusions can be drawn. Best performing SVM models were obtained when 250−500 active and 500−1000 randomly selected inactive training instances were used. Moreover, as long as ∼50 known active compounds are available for training, increasing numbers of 500−1000 randomly selected negative training examples improve and stabilize model performance when class weights are taken into consideration, which provides a clear guideline for virtual compound screening.

Finally, we note that large numbers of active compounds may not always be available for training. However, since SVM classification and ranking models do not take compound potency as a parameter into account, in contrast to support vector regression, large numbers of hits often obtained from confirmatory screening assays might be readily used for SVM model building.

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

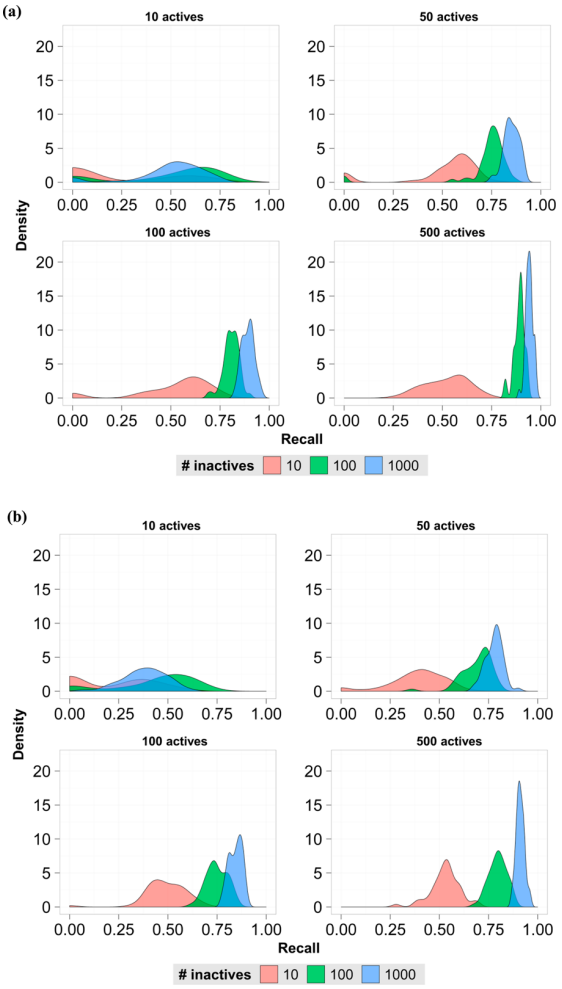


Figure 4. Influence of training set composition and size on recall rates. Density estimates obtained from the distribution of recall rates over 50 trials are presented for training sets of varying size and composition. For a constant number of 10−500 active training compounds, recall distributions are shown for 10 (pink), 100 (green), and 1000 (blue) inactive training compounds: (a) melanocortin receptor 4 ligands, (b) thrombin inhibitors.

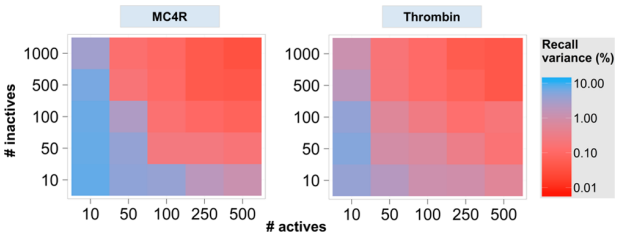


Figure 5. Influence of training set composition and size on recall variance. Heat map representations show variance of recall rates over 50 independent trials (using a two-color gradient) for training sets of varying composition and size: (left) melanocortin receptor 4 (MC4R) ligands, (right) thrombin inhibitors.

AUTHOR INFORMATION

Corresponding Author

*Tel.: +49-228-2699-306. Fax: +49-228-2699-341. E-mail: bajorath@bit.uni-bonn.de.

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 676434, "Big Data in Chemistry" ("BIGCHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) are responsible for any use that may be made of the information it contains.

ABBREVIATIONS

AUC, area under receiver operating characteristic curve; BA, balanced accuracy; ECFP, extended connectivity fingerprint; MC4R, melanocortin receptor 4; RBF, radial basis function; SVM, support vector machine

Table 2. Classification Performance[a]

	10 actives and 100 inactives				500 actives and 1000 inactives			
accession no.	recall μ	recall σ	BA (%) μ	BA (%) σ	recall μ	recall σ	BA (%) μ	BA (%) σ
P00734	0.433	0.211	79.3	5.1	0.911	0.021	98.8	0.6
P00918	0.388	0.219	87.2	3.7	0.770	0.036	97.0	0.9
P21917	0.288	0.164	80.9	5.9	0.744	0.045	96.9	1.1
P41146	0.455	0.163	80.9	6.3	0.924	0.018	99.4	0.3
P00742	0.236	0.138	72.4	5.7	0.872	0.027	98.5	0.6
P29275	0.407	0.226	81.5	4.5	0.820	0.030	97.0	1.1
P32245	0.486	0.276	85.6	4.5	0.942	0.018	99.0	0.5
Q9H3N8	0.440	0.233	84.3	4.5	0.888	0.030	98.4	0.7
Q99705	0.349	0.171	78.7	6.9	0.860	0.046	98.2	0.7
Q9Y5Y4	0.562	0.206	83.7	4.8	0.965	0.013	99.3	0.6
global performance	0.405	0.200	81.4	5.2	0.870	0.028	98.2	0.7

[a]Reported are the mean (μ) and standard deviation (σ) of recall of active compounds and balanced accuracy after 50 independent trials for differently composed training sets: "10 active and 100 inactive compounds" (low performance) and "500 active and 1000 inactive compounds" (high performance). Results are shown for 10 compound classes, referred by accession no., according to Table 1. In addition, global performance over all classes is reported.

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

REFERENCES

(1) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.
(2) Burges, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discovery* **1998**, *2*, 121−167.
(3) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Comput. Chem.* **2001**, *26*, 5−14.
(4) Geppert, H.; Vogt, M.; Bajorath, J. Current Trends in Ligand-Based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation. *J. Chem. Inf. Model.* **2010**, *50*, 205−216.
(5) Heikamp, K.; Bajorath, J. Support Vector Machines for Drug Discovery. *Expert Opin. Drug Discovery* **2014**, *9*, 93−104.
(6) Heikamp, K.; Bajorath, J. Comparison of Inactive and Randomly Selected Compounds as Negative Training Examples in Support Vector Machine-Based Virtual Screening. *J. Chem. Inf. Model.* **2013**, *53*, 1595−1601.
(7) Smusz, S.; Kurczab, R.; Bojarski, A. J. The Influence of the Inactives Subset Generation on the Performance of Machine Learning Methods. *J. Cheminf.* **2013**, *5*, 17.
(8) Kurczab, R.; Smusz, S.; Bojarski, A. J. The Influence of Negative Training Set Size on Machine Learning-Based Virtual Screening. *J. Cheminf.* **2014**, *6*, 32.
(9) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.
(10) Japkowicz, N. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. *AAAI Workshop on Learning from Imbalanced Data Sets*, 2000; Vol. *68*, pp 10−15.
(11) Japkowicz, N.; Stephen, S. The Class Imbalance Problem: A Systematic Study. *Intell. Data Anal.* **2002**, *6*, 429−449.
(12) Stone, M. Cross-Validatory Choice and Assessment of Statistical Predictions. *J. R. Stat. Soc. Series B Stat. Methodol.* **1974**, *36*, 111−147.
(13) Geppert, H.; Horváth, T.; Gärtner, T.; Wrobel, S.; Bajorath, J. Support-Vector-Machine-Based Ranking Significantly Improves the Effectiveness of Similarity Searching Using 2D Fingerprints and Multiple Reference Compounds. *J. Chem. Inf. Model.* **2008**, *48*, 742−746.
(14) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*; Pittsburgh, Pennsylvania; ACM: New York, 1992; pp 144−152.
(15) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Netw.* **2005**, *18*, 1093−1110.
(16) Morik, K.; Brockhausen, P.; Joachims, T. Combining Statistical Learning with a Knowledge-based Approach−A Case Study in Intensive Care Monitoring. In *Proceedings of the 16th International Conference on Machine Learning (ICML-99)*; Morgan Kaufmann: Burlington, MA, 1999.
(17) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; et al. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, 1083−1090.
(18) *MACCS Structural keys*; Accelrys: San Diego, CA, USA, 2006.
(19) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(20) Dodge, Y. *The Concise Encyclopedia of Statistics*; Springer: New York, 2008.
(21) Brown, M. B.; Forsythe, A. B. Robust Tests for the Equality of Variances. *J. Am. Stat. Assoc.* **1974**, *69*, 364−367.
(22) Shaffer, J. P. Multiple Hypothesis Testing. *Annu. Rev. Psychol.* **1995**, *46*, 561−584.
(23) *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria.
(24) Karatzoglou, A.; Smola, A.; Hornik, K.; Zeileis, A. kernlab: An S4 Statistical Package for Kernel Methods in R. *J. Stat. Softw.* **2004**, *11*, 1−20.
(25) Amari, S. I.; Wu, S. Improving Support Vector Machine Classifiers by Modifying Kernel Functions. *Neural Netw.* **1999**, *12*, 783−789.
(26) Alvarsson, J.; Eklund, M.; Engkvist, O.; Spjuth, O.; Carlsson, L.; Wikberg, J. E.; Noeske, T. Ligand-Based Target Prediction with Signature Fingerprints. *J. Chem. Inf. Model.* **2014**, *54*, 2647−2653.

NOTE ADDED AFTER ASAP PUBLICATION

This article was published ASAP on April 10, 2017, with an error in the formula on page B, left column, second paragraph. The corrected version was published ASAP on April 11, 2017.

DOI: 10.1021/acs.jcim.7b00088
J. Chem. Inf. Model. 2017, 57, 710−716

Summary

SVM models were generated for individual targets using distinct amounts and proportions of positive and negative training compounds. Both SVM-based classification and ranking were highly influenced by the numbers of training data. Models became consistently more accurate and robust with increasing training set size. Data imbalance, which is a well-known caveat in ML, did not reduce model performance. Investigated compound sets required a minimum of $\sim$50 actives for training and errors in the minority class (actives) needed to be penalized during the learning process. Under these conditions, increasing the inactive training examples resulted in more stable models that showed consistent high performance regardless of the training set partition. This study has identified the nature of training data as a key factor for successful single-target activity predictions and established guidelines for SVM modeling in activity prediction.

In the following chapter, MT-DNN and ST models are compared at varying density of activity annotations for the prediction of profiling matrices.

Chapter 5

Relative Performance of Multitask Deep Learning and Random Forest Classification on the Basis of Varying Amounts of Training Data

Introduction

As shown in the previous chapter, the training set size and composition has a strong influence on ST-SVM model performance. However, it is not yet understood how varying amounts of training data might affect ST and MT methods, especially for comparative predictions. In *Chapter 2*, different methods and strategies were investigated for the modeling of profiling matrices where MT-DNN did not provide a learning advantage over ST-RF. In this chapter, the availability of all compound-target interaction annotations is hypothesized as a factor influencing the limited performance difference between alternative methods. Thus, the effect of training data sparseness on method relative performance is addressed by training models on matrices of increasing data density.

ACS OMEGA

Article

Cite This: ACS Omega 2018, 3, 12033−12040

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

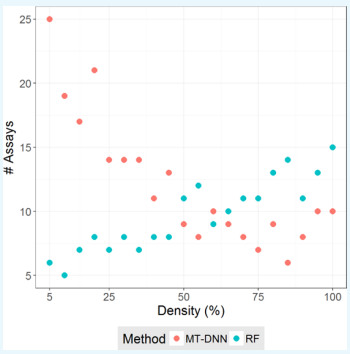
Prediction of Compound Profiling Matrices, Part II: Relative Performance of Multitask Deep Learning and Random Forest Classification on the Basis of Varying Amounts of Training Data

Raquel Rodríguez-Pérez[†,‡] and Jürgen Bajorath*[,†]

[†]Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Endenicher Allee 19c, D-53115 Bonn, Germany

[‡]Department of Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, Birkendorfer Str. 65, 88397 Biberach/Riß, Germany

ABSTRACT: Currently, there is a high level of interest in deep learning and multitask learning in many scientific fields including the life sciences and chemistry. Herein, we investigate the performance of multitask deep neural networks (MT-DNNs) compared to random forest (RF) classification, a standard method in machine learning, in predicting compound profiling experiments. Predictions were carried out on a large profiling matrix extracted from biological screening data. For model building, submatrices with varying data density of 5−100% were generated to investigate the influence of data sparseness on prediction performance. MT-DNN models were directly compared to RF models, and control calculations were also carried out using single-task DNNs (ST-DNNs). On the basis of compound recall, the performance of ST-DNN was consistently lower than that of the other methods. Compared to RF, MT-DNN models only yielded better prediction performance for individual assays in the profiling matrix when training data were very sparse. However, when the matrix density increased to at least 25−45%, per-assay RF models met or partly exceeded the prediction performance of MT-DNN models. When the average performances of RF and MT-DNN over the grid of all targets were compared, MT-DNN was slightly superior to RF, which was a likely consequence of multitask learning. Overall, there was no consistent advantage of MT-DNN over standard RF classification in predicting the results of compound profiling assays under varying conditions. In the presence of very sparse training data, prediction performance was limited. Under these challenging conditions, MT-DNN was the preferred approach. When more training data became available and prediction performance increased, RF performance was not inferior to MT-DNN.



1. INTRODUCTION

Recently, there has been increasing interest in machine learning (ML) and, especially, deep learning (DL) in many areas of science including pharmaceutical research.[1−3] In ML, one can distinguish between single-task (ST) and multitask (MT) learning. MT learning is based on the idea that the predictive performance of a given task can be improved by using the data available for related tasks.[4] In the context of compound activity prediction, which is a core task in computational medicinal chemistry, this principle implies that some structural features and/or molecular properties should be common to active compounds, regardless of their targets. This "basis set" of activity-relevant features would then be complemented by others to yield target-specific biological activities. Hence, bioactivity data from various assays might be considered to predict activities in a given assay on the basis of shared activity determinants, a key assumption underlying MT learning. By contrast, in ST learning, one trains models on the basis of compounds that were active or inactive in an individual assay in order to predict the potential activities of test compounds.

For MT learning, deep neural network (DNN) architectures (MT-DNNs) have become very popular,[2,3] raising expectations that they might yield further improved predictive performance compared to standard ST−ML approaches.[2,3,5] A frequent reasoning is that MT-DNNs make explicit use of more—and more diverse—training data than ST−ML approaches, which further expands the knowledge base for predictions. For example, Ramsundar et al. compared the performance of MT-DNN with different architectures, ST-DNN, and random forest (RF) predictions on four data sets (Kaggle, Factors, Kinase, and UV). Their results suggested that MT models offered improvements over RF calculations for correlated tasks.[3] However, the effect of training matrix density was not explored. Xu et al. compared the performance of ST-DNNs and MT-DNNs for different quantitative structure−activity relationship prediction tasks.[5] Their results indicated that the prediction performance and relative performance of ST-DNNs and MT-DNNs varied greatly across data sets

Received: July 17, 2018
Accepted: September 12, 2018
Published: September 27, 2018

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

containing either on-target potency values or off-target absorption, distribution, metabolism, and excretion properties. Furthermore, Xu et al. concluded that MT-DNN only outperformed ST-DNN when test compounds showed structural similarity and activity that correlated with training set instances from other tasks.[5] Recently, attempts have also been made to predict experimental compound profiling matrices.[6] Such matrices are obtained by screening a compound collection in different assays against closely related or diverse targets and yield activity profiles of test compounds. Importantly, the composition of such matrices is highly unbalanced because the majority of compounds are usually inactive across assays (otherwise, specific biological activities would not exist). In the first investigation,[6] ST and MT models were derived for individual assays in matrices to predict active compounds. Under conditions of experimental data imbalance, prediction performance using different ML approaches was overall reasonable and DNNs did not further increase the performance over RF or support vector machine (SVM) classifiers.[6]

General reasons for varying MT-DNN performance might include, for example, the high complexity of MT-DNN hyperparameter optimization and lack of transparence and/or the nature of training data that is available.[7,8] For example, Rodríguez-Pérez et al. have shown that activity prediction on the basis of ST-SVM classification and ranking became more accurate and stable with increasing numbers of available training instances and that a lower-bound threshold for active training examples was required.[8] In addition, a recent study by de la Vega de León et al. investigated the effects of missing data on the performance of MT methods.[9] In particular, the authors explored the performance of MT-DNN and Macau (Bayesian factorization) methods at different percentages of missing data. A minimum number of training instances was required to generate effective models, but the predictive ability saturated when increasing amounts of data were added.[9] Furthermore, Reker et al. have shown that only small subsets of ligand−target interaction matrices were required for ML modeling to reach upper limits of predictive performance.[10] In this case, RF models were built for predicting interacting versus noninteracting ligand−protein pairs from concatenated molecular and protein descriptors.[10]

Taken together, the studies discussed above have revealed a significant influence of training set size on the quality of both ST− and MT−ML models. However, the influence of training data sparseness on comparative ST− and MT−ML predictions remains to be investigated. Our current study was designed to address this issue by further extending previous work on the modeling of compound profiling matrices,[6] which is a prediction task of high relevance for biological screening and medicinal chemistry. Herein, a large compound profiling matrix combining different screening assays was used to derive submatrices of systematically increasing density for the training of RF, MT-DNN, and ST-DNN models that were then used to predict the activity profile of test compounds. Thereby, the relative performance of predictions using methods of different computational complexity on training matrices of stepwise increasing data density was investigated, thus directly addressing the issue of training data sparseness for comparative prediction of profiling results. The study design and results of our investigation are presented in the following.

2. RESULTS AND DISCUSSION

2.1. Study Design. *2.1.1. Focusing on Profiling Matrices.* Compound profiling matrices from biological screening represent challenging test cases for ML because of the experimental assay variance and, more importantly, inherent data imbalance. This is the case because most screening compounds are inactive in given assays, which typically yield on the order of ∼0.1−1% active compounds (hits).[11] Previously, we have investigated a variety of ML approaches for predicting the experimental results of assays forming complete or nearly complete matrices using the largest possible amount of training data on a per-assay basis.[6] In a complete (100% dense) matrix, all cells are filled with experimental observations. Matrices of decreasing density have increasing amounts of missing data points ("empty" cells). Here, we change the analysis scheme and attempt assay predictions by systematically deriving submatrices of varying density for training, thereby directly assessing the influence of data sparseness on the model quality.

2.1.2. Matrices of Varying Density. From a large profiling matrix comprising more than 140 000 compounds tested in 53 assays (with 0.8% actives), different series of matrices with stepwise increasing data density were extracted, covering the range of 5−100% density, with increments of 5% per step. Further details are provided in the Materials and Methods section. Hence, 20 matrices with varying density levels were obtained. Figure 1 shows the distribution of the number of active compounds per assay for five exemplary matrices with different densities of 5, 25, 50, 75, and 100%, respectively. The figure illustrates that increasing data density correlated with increasing numbers of active compounds available for training.

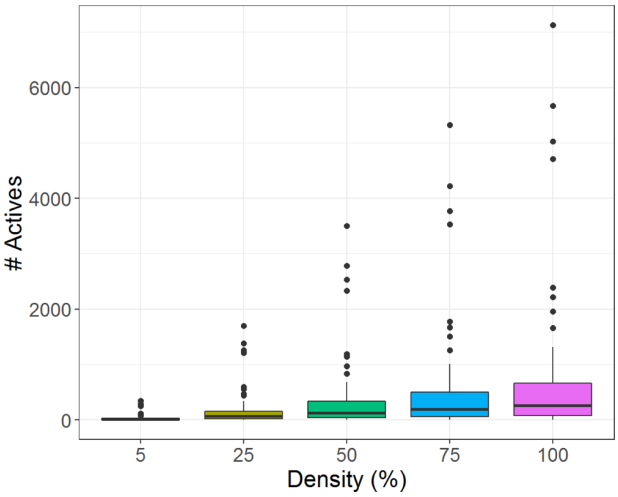


Figure 1. Active compounds per assay. Distributions of the number (#) of active compounds per assay are reported in boxplots for five different matrix density levels. Black points represent outliers.

2.1.3. Training and Predictions. For each of the 20 matrices with increasing density, ML models were derived at each density level. The resulting models were then used to predict active compounds. For ST predictions, an individual model was built for each assay (target) to predict active compounds on a per-assay basis. Individual predictions were then combined. For MT predictions, multioutput models were derived for all assays at each density level to predict the

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

complete activity profile of a compound. The resulting ST and MT models were used to predict a constant test set comprising 25% of the original profiling matrix that was excluded from training.

2.1.4. Selected Methods. As an ST−ML approach, RF was selected. This choice was motivated by the results of our previous ST matrix predictions where RF was the overall best approach, achieving slightly better performance than SVMs and ST-DNNs.[6] As an MT−ML method, MT-DNN was chosen, which represents the currently most complex MT approach. Thus, RF and MT-DNN essentially delineate opposite ends of the ML spectrum ranging from methods of low to high computational complexity and an increasing "black box" character. As a control, ST-DNN models were also generated and evaluated.

In the following, the results of our systematic activity predictions using RF and MT-DNN models trained at different density levels are presented and compared. The results were averaged over three independent trials.

2.2. Influence of Matrix Density on Prediction Performance. We first investigated how training sample sizes influenced the predictive ability of ST models based upon data from only one assay or MT models based upon data from all assays. Therefore, a pairwise comparison of ST or MT models at different density levels was carried out using the area under the receiver operating characteristic (ROC) curve (AUC) as a figure of merit. For a given assay, the AUC difference at two density levels was required to exceed 2% to classify one prediction to be superior to another. The training matrix yielding the best (worst) performing model was considered to be of *superior* (*inferior*) density. Figure 2a,b reports the results for RF and MT-DNN, respectively. The number of assays for which a model trained with a given matrix density provides better results compared to another matrix density is reported. In addition to the pairwise comparison shown in the heatmap, the panel on the right reports a cross-density comparison for the same method. For both methods, models trained at higher density levels produced better predictions on a per-assay basis than the models trained at lower density levels, as clearly revealed by the heatmap representations. Thus, consistent with earlier observations, increasing numbers of positive training instances resulted in increasing prediction performance, here for both ST and MT models. The separation between predictions with models trained at higher or lower density was even more extensive for RF than MT-DNN, as also indicated by the distribution of superior assay counts in Figure 2. Hence, RF models were overall more affected by missing data than MT-DNN models.

2.3. Method Comparison. Next, the performance of RF and MT-DNN was compared at different density levels.

2.3.1. Relative Performance for Individual Tasks. Prediction performance was first compared on a per-assay basis using AUC and Matthew's correlation coefficient (MCC). A model was considered superior if it achieved at least 2% better performance than its counterpart. This criterion was used as a disjunctive requirement for the AUC and MCC measures. Then, the number of individual assays in which a method was superior to another was separately calculated for both figures of merit. Figure 3 reports the average number of assays for three independent trials. Figure 3a shows the mean number of assays with larger AUC values for a given method at varying density levels. MT-DNN was clearly superior to RF when very sparse matrices were used for training. However, at increasing density

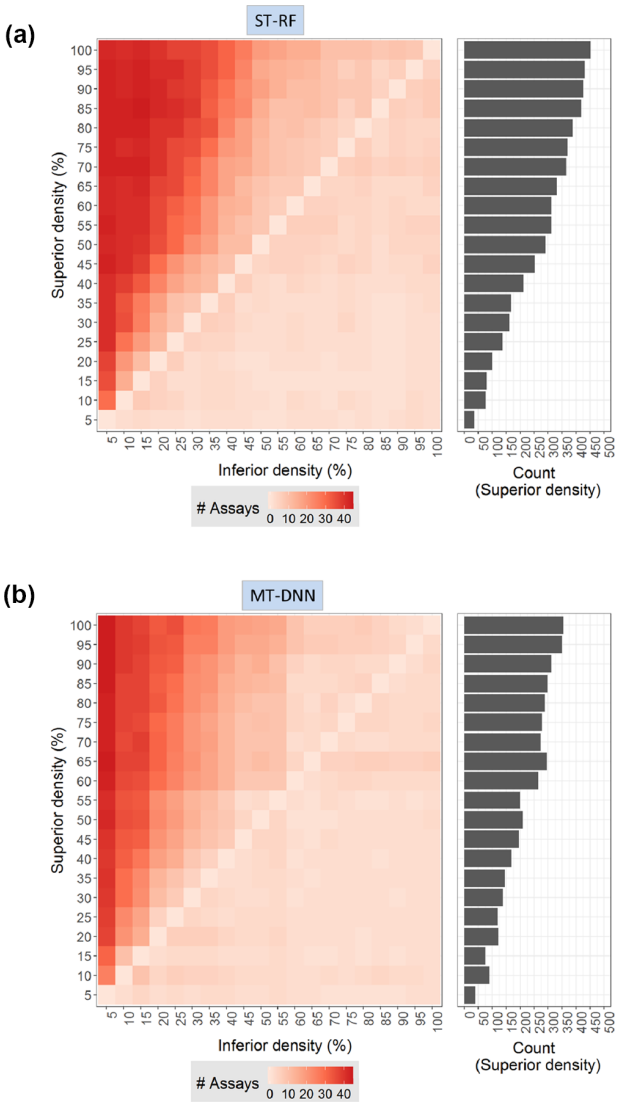


Figure 2. Prediction performance at different matrix densities. Heatmaps record the average number of assays for which larger AUC values were obtained at a given (superior) matrix density (*y*-axis) compared to another (inferior) density (*x*-axis). On the right, bar graphs report the number of assays (count) at a given density level for which better prediction performance was achieved than at any other density for the same method. (a) RF and (b) MT-DNN.

levels, performance differences became smaller, and at a density level of 50% or greater, the performance of RF began to meet and then slightly exceed the performance of MT-DNN. Figure 3b reports the corresponding comparison on the basis of MCC calculations. In this case, MT-DNN models produced better predictions at low density levels of up to 25%. At further increasing density, however, RF models were clearly superior to MT-DNN. Thus, on the basis of the AUC and MCC performance measures, similar trends were observed on a per-assay basis, with MT-DNN models yielding better prediction performance for training on very sparse matrices and RF models having better prediction performance at increasing density levels, especially when evaluated on the basis of MCC calculations. At high density levels, that is, in the presence of large amounts of training data, RF models were

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

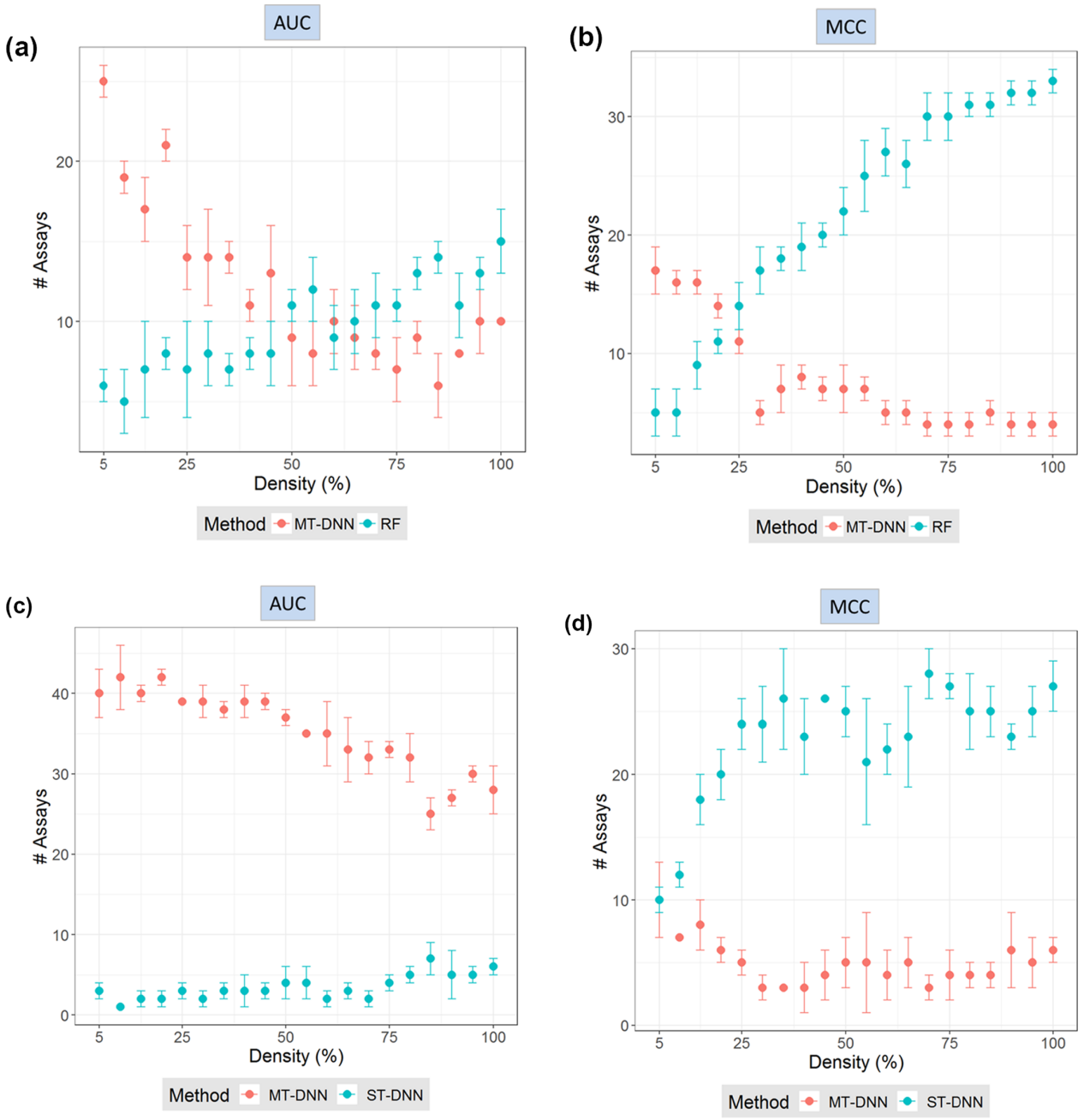


Figure 3. Per-assay comparison of prediction performance using different methods. For different trials covering all matrix density levels, the mean (dot) and standard deviation (error bar) of the number of assays are given for which one method achieved higher prediction performance than the other on the basis of different measures. RF, MT-DNN, and ST-DNN models were compared. (a) MT-DNN vs RF on the basis of AUC, (b) MT-DNN vs RF; MCC, (c) MT-DNN vs ST-DNN; AUC, and (d) MT-DNN vs ST-DNN; MCC.

superior on a per-assay basis to the much more complex MT-DNN models.

To provide additional control calculations, ST-DNN models were also generated. Figure 3c compares ST- and MT-DNN models on the basis of AUC values. MT-DNN models outperformed ST-DNN models in most assays at varying density levels. ST-DNN models only yielded better AUC values in a few cases. At decreasing matrix density, performance differences between MT- and ST-DNN increased, and MT-DNN was progressively superior. Figure 3d shows the results of MCC calculations. Here, ST-DNN models yielded larger MCC values for more assays than MT-DNN models. However, for very sparse training matrices, the relative performances of both methods became comparable.

2.3.2. Global Prediction Performance. Figure 4a shows the mean AUC, balanced accuracy (BA), and MCC values over all assays at varying density levels. Values of different performance measures are reported in Table 1. Different from the results obtained for individual assays, on average, predictions were slightly superior for MT-DNN compared to RF when assessed on the basis of AUC and clearly superior on the basis of BA calculations. However, on the basis of MCC calculations, the

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

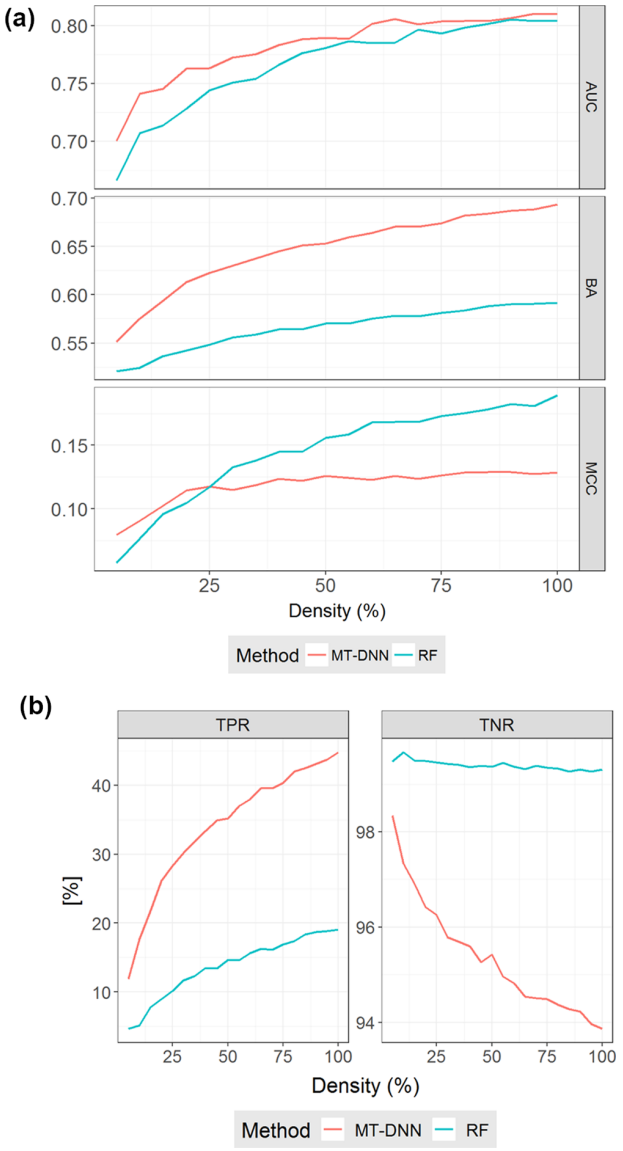


Figure 4. Global prediction performance using different methods. For different trials covering all matrix density levels, the mean prediction performance over all assays is compared for MT-DNN and RF using different measures. (a) AUC (top), BA (middle), and MCC (bottom), (b) TPR (right), and TNR (left).

global prediction was only slightly better for MT-DNN models at very low density levels of up to 25%. Then, the prediction performance of RF models gradually exceeded the performance of MT-DNN models, consistent with the results in Figure 3b. Hence, Figure 4a shows that different performance measures produced different results. As a consensus, we would conclude that average results over all assays were slightly better for MT-DNN than RF.

To better understand apparent differences resulting from the application of alternative performance measures, confusion matrices were generated at different density levels using mean values. Rates derived from raw counts of true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs) were calculated. Figure 4b shows the TP rate (TPR) and TN rate (TNR), which are defined as follows: TPR = TP/(TP + FN) and TNR = TN/(TN + FP). Therefore, TPR and TNR are related to FN rates (FNR) and FP rates (FPR), respectively. TPR and FNR displayed the same tendency for RF and MT-DNN. At increasing density, TPR increased and FNR decreased. However, for MT-DNN, FPR increased and TNR decreased at increasing density levels, whereas they remained essentially constant for RF across all levels. Thus, MT-DNN predicted more FPs than RF at increasing density. We note that the constantly used test set contained a mean of 35 523 inactive and only 305 active compounds per target, given the inherent data imbalance. Consequently, figures of merit that use absolute values such as MCC are strongly affected by the different magnitudes of the numbers of active and inactive compounds. Conversely, other measures relying on proportions only yield small differences, which correspond, however, to large differences in the absolute number of errors.

On the basis of MCC calculations, MT-DNN model performance was clearly inferior to RF, except at lower density levels, when the number of FPs and TNs decreased and increased, respectively. On the other hand, the model performance assessed by BA taking only the TPR and TNR into account was superior for MT-DNN, given that the TPR was consistently higher for MT-DNN and differences in TNR were comparably small. These aspects must be taken into consideration when judging relative prediction performance on imbalanced data sets using alternative figures of merit.

ST-DNN was also included in the global comparison as a control. On the basis of AUC values, ST-DNN performed consistently worse than the other two methods. In addition, ST-DNN models produced BA values falling in between those of RF and MT-DNN and MCC values that were overall comparable to RF.

The consensus view emerging from the results comparing MT-DNN and RF shown in Figures 3 and 4 was that MT-DNN was only superior to RF when models were trained on the basis of very sparse matrices. When examining the relative prediction performance (Figure 3), MT-DNN models only

Table 1. Evaluation of Predictions Applying Different Performance Measures[a]

	AUC		BA		MCC		TPR		TNR	
matrix density (%)	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF
5	0.700	0.666	0.551	0.521	0.080	0.058	11.9	4.6	98.3	99.5
25	0.763	0.744	0.623	0.548	0.117	0.117	28.3	10.2	96.3	99.5
50	0.790	0.781	0.653	0.570	0.126	0.156	35.2	14.7	95.4	99.4
75	0.803	0.793	0.674	0.581	0.126	0.173	40.3	16.9	94.5	99.3
100	0.810	0.804	0.693	0.591	0.128	0.190	44.8	19.1	93.9	99.3

[a]Reported are mean AUC, BA, and MCC values for global predictions using MT-DNN and RF models trained at varying matrix density levels. In addition, mean TPR and TNR values are given.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

displayed superior performance to RF models at training matrix density levels of up to 25−45%, depending on the performance measures that were applied. By contrast, at increasing matrix density, RF calculations often met or exceeded the prediction performance of MT-DNN at the level of individual assays. Global prediction results (Figure 4) also showed that when enough training data were available, RF models were at least as good as MT-DNN models. Only global BA values were consistently higher for MT-DNN, but for the remaining performance measures (AUC, MCC), MT learning only provided a notable advantage at low matrix density levels.

2.4. Concluding Discussion. In this work, we have systematically explored the effects of using varying amounts of training data on MT-DNN and RF modeling. As a prediction task representing experimental results, a large compound profiling matrix was selected. The analysis was facilitated by generating assay submatrices of varying density for model derivation. The resulting models were then compared on the basis of a consistently used test submatrix of 100% density. There was no significant global correlation between prediction tasks. Differences in the performance of (low-complexity) RF and (high-complexity) MT-DNN models were observed at different density levels.

When trained on very sparse matrices, MT-DNN models yielded better prediction performance than RF models. However, when the density increased to 25−45%, per-assay RF models met or slightly exceeded the prediction performance of MT-DNN models. Thus, compared to a RF, a standard ML classifier, MT-DNN models only provided a learning advantage for individual assays when training data were very limited. However, when predictions were averaged over all assays, MT-DNN was the overall superior approach, albeit by a confined margin, depending on the applied performance measures. This observation reflected the presence of more stable predictions as a likely consequence of MT learning. On the basis of AUC values, ST-DNN was consistently inferior to MT-DNN and RF but produced higher MCC values than MT-DNN for matrices of increasing density. In all instances, performance assessment yielded partly different results, depending on the measures that were used, emphasizing the need to consider alternative performance measures in ML.

Taken together, the results of our analysis show that there was no consistent advantage of MT-DNNs over RF in predicting profiling assay results, as one might have anticipated, given high expectations often associated with MT DL. These findings should balance such expectations, at least for applications of DL in compound screening. However, they are also encouraging from the point of view that reasonable prediction performance was also achieved on a complicated prediction task with a standard ML classifier of much lower complexity than DNN architectures. Clearly, under most challenging conditions of data sparseness, when prediction performance was limited, MT-DNN was the superior approach. When increasing amounts of training data became available, and the model quality generally improved, the performance of MT-DNN and RF was comparable.

Taken together, our findings also suggest that MT-DNN might be preferred over standard classification methods such as RF in special situations, for example, when the main objective is modeling a single task (activity) and only very little training data are available for this task, but extensive data are available for related (correlated) tasks (such as similar activities). In addition, MT-DNN might be an approach of choice when the main objective is improving global prediction performance over multiple screens, and only sparse training matrices are available.

In future work, additional prediction tasks in chemistry and other challenging prediction conditions should be explored to further evaluate potentially significant advantages of DL and MT learning over standard ML approaches.

3. MATERIALS AND METHODS

3.1. Assay Data. A large compound profiling matrix was algorithmically extracted[12] from PubChem confirmatory assays as described previously[6] and provided the basis for our analysis. This matrix consisted of 143 310 compounds tested in 53 assays (covering a diverse range of 53 unique target proteins).[6] In the matrix, activity versus inactivity of compounds in assays was recorded in a binary format (i.e., 1 vs 0). The matrix density of experimental observations was 96.4%. As reported in Table 2, the majority of screened compounds (77%) were consistently inactive in all assays, 13% of the compounds had single-target activity, and 10% had multitarget activity. The resulting global proportion of matrix cells containing activity annotations was 0.8%. As reported previously,[6] the intra- and interassay similarity of active matrix compounds was generally low.

Table 2. Matrix Compounds with Different Activity Status[a]

activity status	number of compounds
consistently inactive	110 272 (77%)
single-target activity	19 054 (13%)
multitarget activity	13 984 (10%)

[a]Reported are the numbers of matrix compounds with different activities across all assays.

3.2. Matrix Modifications. For computational modeling, the matrix was completed (100% density) by conventional zero filling,[13] that is, missing experimental data (3.6%) were compensated for by inactivity annotations. The complete matrix was then randomly divided into training (75%) and test data (25%). The test set submatrix was complete (100% density). By contrast, training sets of varying density were created ranging from 5 to 100% density, with increments of 5%. To these ends, 95% of the compound-assay annotations were randomly removed, and assay data were added back in 5% increments, yielding cumulatively built training sets of stepwise increasing density.

3.3. Machine Learning. Using a consistent molecular representation, two distinct ML approaches of different designs and computational complexity were investigated including (ST-)RF and MT-DNN. As a control, ST-DNN calculations were carried out.

3.3.1. Molecular Representation. The folded (1024-bit) version of the extended connectivity fingerprint with bond diameter 4 (ECFP4) was used as a molecular representation.[14] ECFP4 was computed using in-house Python scripts based upon the OEChem Toolkit.[15]

3.3.2. Calculation Protocol. For each matrix density level, RF and MT-DNN models were trained and used to predict the same test data. Three independent trials with different random seeds were carried out for training sets covering all density levels, as detailed above. In each trial, RF and MT-DNN models were built for individual assays using the same cumulative training sets and compared. The use of different

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

random seeds for modeling modified the initialization of MT-DNN and cross-validation partitions of RF models.

Models were only built for assays for which the training matrices and the test matrix consistently contained active compounds. The different training sets included active compounds from all 53 assays, whereas the test set was found to contain active compounds from 47 assays. Thus, RF and MT-DNN models were ultimately built for 47 assays (targets).

Figure 5 shows the distribution of pairwise Pearson correlation coefficients (r) between learning tasks encoded

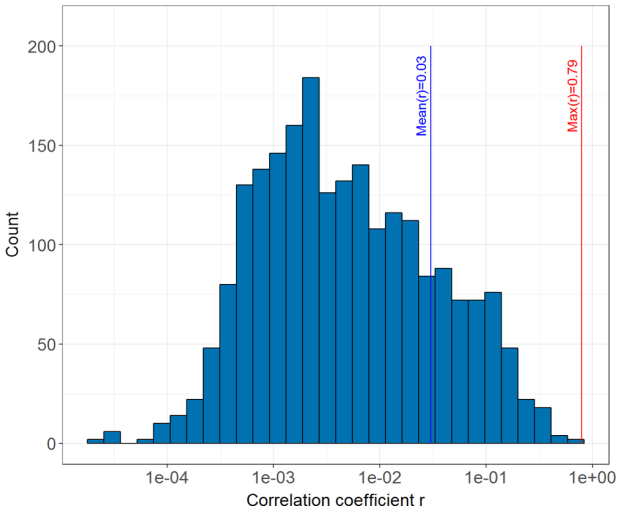


Figure 5. Correlation between learning tasks. The histogram shows the distribution of pairwise Pearson correlation coefficients (r) between all learning tasks (training data) on a logarithmic x-scale.

by the matrix. The maximum r value was 0.79 and the mean r value 0.03, which indicated very low global correlation between tasks (while significant correlation between tasks typically supports transfer and MT learning).

3.3.3. Random Forest. The RF approach utilizes an ensemble of decision trees that are built with different subsets of samples by bootstrapping.[16] Variance is reduced by training decision trees using different subsets of the training set. Moreover, a random sample of features is considered during node splitting, which avoids the presence of correlated trees because of feature dominance.[16] In this study, the *scikit-learn* implementation of RF was used.[17] The number of trees was set at 100, and two hyperparameters were optimized using twofold cross-validation including the number of randomly selected features available at each bifurcation (*max_features*) and the minimum number of samples required to reach a leaf node (*min_samples_leaf*). Cross-validation optimization was independently carried out on a per-assay basis such that different optimum hyperparameters could be derived for each RF model. Tested values for *max_features* included the total number of features, the square root, and the logarithm to base two of the number of features. In addition, for *min_samples_leaf*, candidate values were 1, 5, and 10. Class weights were set according to the ratio of samples from each training label (i.e., active vs inactive) such that errors in the minority class were preferentially penalized.[7] Default values were used for all remaining hyperparameters.[17]

3.3.4. Multitask Deep Neural Networks. Feed-forward DNNs learn a function that approximates the input values to an output (class) without backward connections or loops within the network architecture.[18,19] DNNs can be used for MT activity predictions by considering multiple nodes in the output layer, yielding MT-DNNs.[19] A DNN is constituted by different layers including an input layer, hidden layers, and an output layer.[20] Each layer contains a number of neurons that assign weights to the values originating from the previous layer, adds them, and passes the sum through an activation function

$$y_k = f\left(\sum_j w_{kj}x_j + b_k\right)$$

Here, y_k is the output and x_j is the input of neuron k, f is the activation function, w_{kj} are the weights connecting neuron k with x_j, and b_k is the so-called bias.[21] Ultimately, the output layer transforms the values of the last hidden layer into the output values (classes). Weights are derived during training by the iterative value modification to obtain the desired output y. Gradient descent is computed using back-propagation to optimize the weights and biases.[20] For weight and bias adjustment, back-propagation required the actual labels (active/inactive) of the training set. For MT-DNN calculations, Keras[22] and TensorFlow[23] Python implementations were used.

For MT-DNNs, many optimization-relevant hyperparameters are available. Because 20 successive density levels and three trials per level were investigated, an exhaustive evaluation of alternative hyperparameter settings was computationally infeasible. Instead, a set of hyperparameters permitting validation loss convergence was chosen for comparison of different density levels, as suggested by previous optimization studies.[6,20] These parameter settings included, first, a pyramidal network architecture with two hidden layers of 2000 and 1000 neurons, respectively. In addition, the rectified linear unit (ReLU) function was chosen as an activation function, except for the output layer, in which the sigmoid function was employed. Furthermore, as an optimization function, stochastic gradient descent (SGD) was used, the batch size was 1024, and the initial learning rate (LR) was set to 0.01 and iteratively decreased when the training loss reached a plateau and remained constant. To avoid overfitting, a fall-out rate of 25% was applied. A total of 800 epochs were computed, and the best resulting model was used for prediction. Class weights were considered. For internal validation, an 80−20% data split was applied. Binary cross-entropy was used as the loss function and the reduction of the LR and the choice of the best model after 800 epochs were based on minimizing this validation loss.

3.3.5. Single-Task Deep Neural Networks. As additional control calculations, ST-DNN models were built and evaluated at the same 20 density levels. Hyperparameter values were set according to previous optimization results.[6] The ST-DNN network architecture included two hidden layers with 2000 and 1000 neurons, respectively. ReLU was the activation function, except for the output layer, which used the softmax function. The optimization function was SGD, the batch size was set to 128, and the LR was set to 0.0001. To avoid overfitting, a drop-out rate of 25% was permitted, and L2-regularization was applied. Furthermore, batch normalization was applied to all layers, and a total of 100 epochs were computed.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

3.4. Performance Measures. The performance of ML models was evaluated using confusion matrices and three different measures including the area under the ROC curve (AUC),[24] MCC,[25] and BA.[26] AUC evaluates the global ranking of test compounds. MCC and BA are defined below

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

$$\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$$

AUTHOR INFORMATION

Corresponding Author

*E-mail: bajorath@bit.uni-bonn.de. Phone: 49-228-7369-100 (J.B.).

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Author Contributions

The study was carried out, and the manuscript was written with contributions of all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 676434, "Big Data in Chemistry" ("BIG-CHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) are responsible for any use that may be made of the information it contains. The authors thank the OpenEye Scientific Software, Inc., for providing a free academic license of the OpenEye toolkit. The authors also thank Nils Weskamp for support and helpful discussions.

REFERENCES

(1) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discovery Today* **2018**, *23*, 1241−1250.

(2) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.

(3) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.

(4) Caruana, R. Multitask Learning. In *Learning to Learn*; Thrun, S., Pratt, L., Eds.; Springer: New York, 1998; pp 95−133.

(5) Xu, Y.; Ma, J.; Liaw, A.; Sheridan, R. P.; Svetnik, V. Demystifying Multitask Deep Neural Networks for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2017**, *57*, 2490−2504.

(6) Rodríguez-Pérez, R.; Miyao, T.; Jasial, S.; Vogt, M.; Bajorath, J. Prediction of Compound Profiling Matrices Using Machine Learning. *ACS Omega* **2018**, *3*, 4713−4723.

(7) Kurczab, R.; Bojarski, A. J. The influence of the negative-positive ratio and screening database size on the performance of machine learning-based virtual screening. *PLoS One* **2017**, *12*, No. e0175410.

(8) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.

(9) de la Vega de León, A.; Chen, B.; Gillet, V. J. Effect of Missing Data on Multitask Prediction Methods. *J. Cheminf.* **2018**, *10*, 26.

(10) Reker, D.; Schneider, P.; Schneider, G.; Brown, J. B. Active Learning for Computational Chemogenomics. *Future Med. Chem.* **2017**, *9*, 381−402.

(11) Zhu, T.; Cao, S.; Su, P.; Patel, R.; Shah, D.; Chokshi, H.; Szukala, R.; Johnson, M.; Hevener, K. E. Hit Identification and Optimization in Virtual Screening: Practical Recommendations Based Upon a Critical Literature Analysis. *J. Med. Chem.* **2014**, *56*, 6560−6572.

(12) Vogt, M.; Jasial, S.; Bajorath, J. Extracting Compound Profiling Matrices from Screening Data. *ACS Omega* **2018**, *3*, 4706−4712.

(13) Tanrikulu, Y.; Kondru, R.; Schneider, G.; So, W. V.; Bitter, H.-M. Missing Value Estimation for Compound-Target Activity Data. *Mol. Inf.* **2010**, *29*, 678−684.

(14) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(15) *OEChem TK version 2.0.0*; OpenEye Scientific Software: Santa Fe, NM, 2015.

(16) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.

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

(18) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3−14.

(19) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, 2016.

(20) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015.

(21) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.

(22) Chollet, F. Keras, version 2.1.3, 2015 https://github.com/keras-team/keras (accessed Jan. 17, 2018).

(23) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M., Yu, Y.; Zheng, X. TensorFlow: A System for Large-scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*: Savannah, GA, 2016.

(24) Bradley, A. P. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recogn.* **1997**, *30*, 1145−1159.

(25) Matthews, B. W. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta* **1975**, *405*, 442−451.

(26) Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; Buhmann, J. M. The Balanced Accuracy and Its Posterior Distribution. *20th International Conference on Pattern Recognition*, 2010.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

Summary

Herein, a dense compound profiling matrix extracted from PubChem BioAssay facilitated the study of the influence of data sparseness on model performance. A systematic analysis of MT-DNN and ST-RF predictions demonstrated a strong influence of profiling matrix density on global and relative methods' performance. No consistent advantage was observed for MT-DNN models compared to ST-RF. Average or global performance estimations using distinct metrics pointed out MT-DNN superiority for the prediction of activity profiles. However, performance for individual targets highlighted MT-DNN as a preferred option only for modeling very sparse data sets. Nevertheless, for increasing matrix density and increasing numbers of compound-target annotations as training data, ST-RF results on individual assays approached and occasionally exceeded MT-DNN predictive ability.

In the previous chapters, ML models of varying complexity have shown high performance in predicting compound activity across distinct test cases. The following chapters cover the work on interpretability of complex model decisions.

Chapter 6

Support Vector Machine Classification and Regression Prioritize Different Structural Features for Binary Compound Activity and Potency Value Prediction

Introduction

SVM is a standard ML method that provides state-of-the-art performance for the prediction of compound activity against biological targets, as also shown in previous chapters. The extension of the SVM algorithm to regression problems (SVR) has also allowed the prediction of potency values or, in other words, the magnitude of activity. Both variants relate structural parts of chemical compounds to changes in activity resulting in qualitative and quantitative SAR models, which typically are non-linear. In this chapter, a systematic analysis of feature relevance in non-linear SVM and SVR models is presented. The aim is determining whether prioritized chemical patterns are stable across independent trials and common in both models. Important features for both methods are extracted and visualized to compare chemical patterns driving the binary activity and potency value predictions.

ACS OMEGA

Article

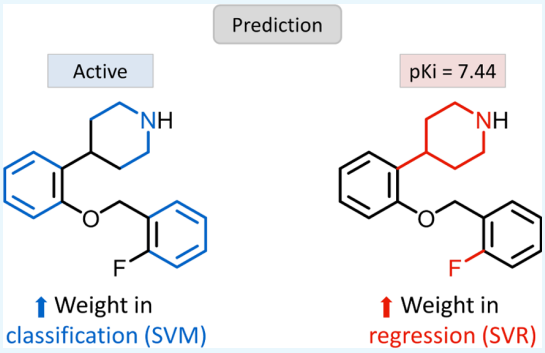
http://pubs.acs.org/journal/acsodf

Support Vector Machine Classification and Regression Prioritize Different Structural Features for Binary Compound Activity and Potency Value Prediction

Raquel Rodríguez-Pérez, Martin Vogt, and Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstr. 2, D-53113 Bonn, Germany

ABSTRACT: In computational chemistry and chemoinformatics, the support vector machine (SVM) algorithm is among the most widely used machine learning methods for the identification of new active compounds. In addition, support vector regression (SVR) has become a preferred approach for modeling nonlinear structure−activity relationships and predicting compound potency values. For the closely related SVM and SVR methods, fingerprints (i.e., bit string or feature set representations of chemical structure and properties) are generally preferred descriptors. Herein, we have compared SVM and SVR calculations for the same compound data sets to evaluate which features are responsible for predictions. On the basis of systematic feature weight analysis, rather surprising results were obtained. Fingerprint features were frequently identified that contributed differently to the corresponding SVM and SVR models. The overlap between feature sets determining the predictive performance of SVM and SVR was only very small. Furthermore, features were identified that had opposite effects on SVM and SVR predictions. Feature weight analysis in combination with feature mapping made it also possible to interpret individual predictions, thus balancing the black box character of SVM/SVR modeling.



1. INTRODUCTION

Supervised machine learning is a preferred approach for the prediction of compound properties including biological activity.[1,2] Among machine learning approaches, support vector machines (SVM) have become increasingly popular.[3−5] The SVM methodology was originally conceived for binary class label prediction of objects[6−8] on the basis of training data. In a given feature space, SVM learning aims to construct a hyperplane to best separate training data with different class labels.[7,8] The hyperplane is derived on the basis of a limited number of training instances, so-called support vectors, to maximize a margin on each side of the plane. If the data are not separable by a hyperplane, the data can be projected into feature spaces of higher dimensionality where linear separation of positive and negative examples might be possible.[7,8] For a given feature space, a successfully derived hyperplane represents a classification model that can then be used to predict the class label of test objects in this space, depending on which side of the hyperplane (i.e., the positive or negative) they fall. In chemoinformatics, binary class label prediction is used for compound classification, for example, to distinguish active from inactive compounds.[3,4] In addition to class label prediction, SVM models can also be used for compound database ranking by calculating their distance from the "active" or "inactive side" of the hyperplane.[9]

Support vector regression (SVR), an extension of the SVM algorithm, has been introduced for predicting numerical property values[10,11] such as compound potency. In SVR, instead of generating a hyperplane for class label prediction, a different function is derived on the basis of training data to predict numerical values. In analogy to SVM, SVR also projects training data with nonlinear structure−activity relationships (SARs) in a given feature space into higher-dimensional space representations where a linear regression function may be derived. In this case, compounds with different potency values are used to fit a regression model that can then be used to predict the potency of new candidate compounds. SVR typically produces statistically accurate regression models when predictions over all potency ranges are analyzed.[5,12] However, SVR also displays the tendency to underpredict highly potent compounds in data sets and hence eliminates activity cliffs from their activity landscape.[12]

In SVM and SVR, mapping into higher-dimensional feature spaces, which is a signature of these algorithms, is accomplished through the use of kernel functions, the so-called "kernel trick".[13] When using nonlinear kernel functions, SVM and SVR can resolve nonlinear SARs in original feature spaces through dimensionality extension. This makes SVR especially attractive for potency prediction because it is not confined to the applicability domain of conventional quantitative SAR analysis

Received: July 27, 2017
Accepted: September 22, 2017
Published: October 4, 2017

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

methods.[14] On the other hand, both SVM and SVR modeling have black box character, meaning that the predictions cannot be directly interpreted in chemical terms. Hence, it is generally difficult to rationalize model performance. Only few attempts have thus far been made to aid in SVM model interpretation in high-dimensional kernel spaces. For example, support vectors with largest contributions to SVM models have been visualized.[15] In addition, descriptor features have been organized in polar coordinate systems according to their contributions to SVM predictions.[16]

To increase model interpretability and reduce the black box character of SVM and SVR, we aimed to identify descriptor features that determine model performance on individual compound data sets. Given the close methodological relationship between SVM and SVR, relevant features of classification and regression models were also compared. Intuitively, one might expect that SVM and SVR would prioritize similar features for a given compound data set because most informative chemical features for predicting whether a compound is active or not might also be relevant for predicting the magnitude of activity. For this purpose, feature weighting and mapping techniques were systematically applied. Feature mapping helped to rationalize the performance of SVM and SVR models.

2. RESULTS AND DISCUSSION

2.1. Global Performance of SVM and SVR Models. A prerequisite for feature weight analysis is the assessment of the prediction accuracy of SVM and SVR models. This is the case because the evaluation of features that contribute to predictions is only meaningful if the underlying models reach a reasonably high-performance level. Figure 1 summarizes the performance of our SVM and SVR models on the 15 activity classes using different figures of merit appropriate for assessing classification and regression calculations. Results are presented for two molecular representations, the MACCS fingerprint and extended connectivity fingerprint with bond diameter 4 (ECFP4). Figure 1a shows that the median F1 scores and the area under the ROC curve (AUC) values of the SVM models were clearly above 0.95 for both MACCS and ECFP4 fingerprints, reflecting accurate classification of active and inactive compounds. Furthermore, recall rates of the active compounds reached a median value of 0.77 for MACCS and 0.94 for ECFP4 among the top 1% of the ranked compounds. These results also reflected the usually observed higher performance of ECFP4 relative to MACCS.

Figure 1b reports the performance of the SVR models across the different activity classes. The median values of mean absolute error (MAE) and mean squared error (MSE) median values were between 0.5 and 0.6, and the median values of the Pearson correlation coefficient (r) between the predicted and observed pK_i values were above 0.7 for MACCS and above 0.8 for ECFP4. In addition, errors of potency predictions were consistently limited to less than 1 order of magnitude. Thus, the SVR model also exhibited an overall reasonable performance.

2.2. Feature Relevance. A second condition for informative feature weight analysis is demonstrating the relevance of individual fingerprint features. Therefore, features were randomly removed from SVM models or in the order of decreasing feature weights, and classification calculations were repeated. Figure 2 shows the results for exemplary activity classes and the MACCS (Figure 2a) and ECFP4 (Figure 2b)

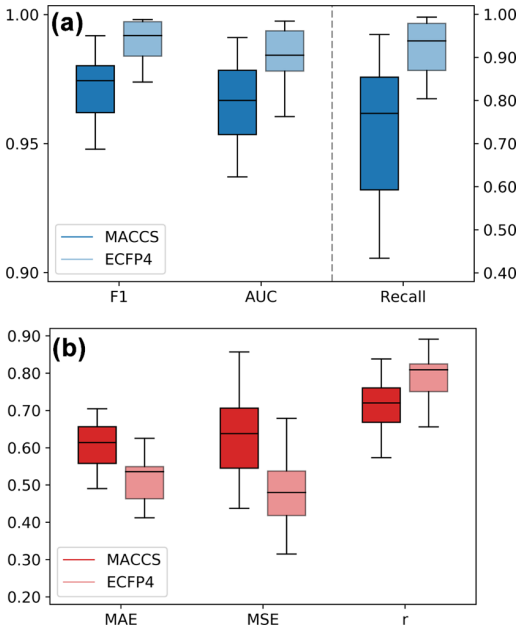


Figure 1. Global performance. Box plots report the prediction accuracy of (a) SVM and (b) SVR calculations over all activity classes and 10 independent trials per class. For SVM calculations, the F1 score, AUC, and recall of active compounds among the top 1% of the ranked test set are reported. For SVR calculations, the MAE and MSE values and the Pearson correlation coefficient (r) for the observed and predicted potency values are given.

fingerprints. For MACCS containing 166 features, both random and weight-based feature removal decreased compound recall and increased MSE values. The magnitude of errors was greater for weight-based feature removal than for random feature removal. For ECFP4 comprising much larger numbers of possible features, random feature removal affected the calculations only marginally, if at all, whereas removal of highly weighted features led to a substantial reduction in compound recall and a gradual increase in MSE values. Thus, as anticipated, removal of features obtaining high weights during model building consistently reduced the model performance.

2.3. Global Feature Weight Analysis. For SVM and SVR models, weights of fingerprint features were systematically determined over 10 independent trials and compared. In some instances, feature weights were consistently high or low over different trials, as further detailed below; in others, they varied depending on the training data. In addition, feature weights generally varied for different activity classes, as expected. Furthermore, it was observed that some individual features were equally important for SVM and SVR for a given class, consistent with their shared methodological framework.

However, a striking finding was that the importance of many features for classification and regression fundamentally differed. Figures 3 and 4 show representative examples for different activity classes and MACCS and ECFP4, respectively. Feature weights were assigned to three different categories (i.e., high, medium, and low), as detailed in the Materials and Methods section. Figures 3a and 4a show examples of MACCS and ECFP4 features, respectively, which had very different weights in SVM and SVR models, including features with consistently—or mostly—low weights in classification and high weights in regression model and vice versa. Thus, many features

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

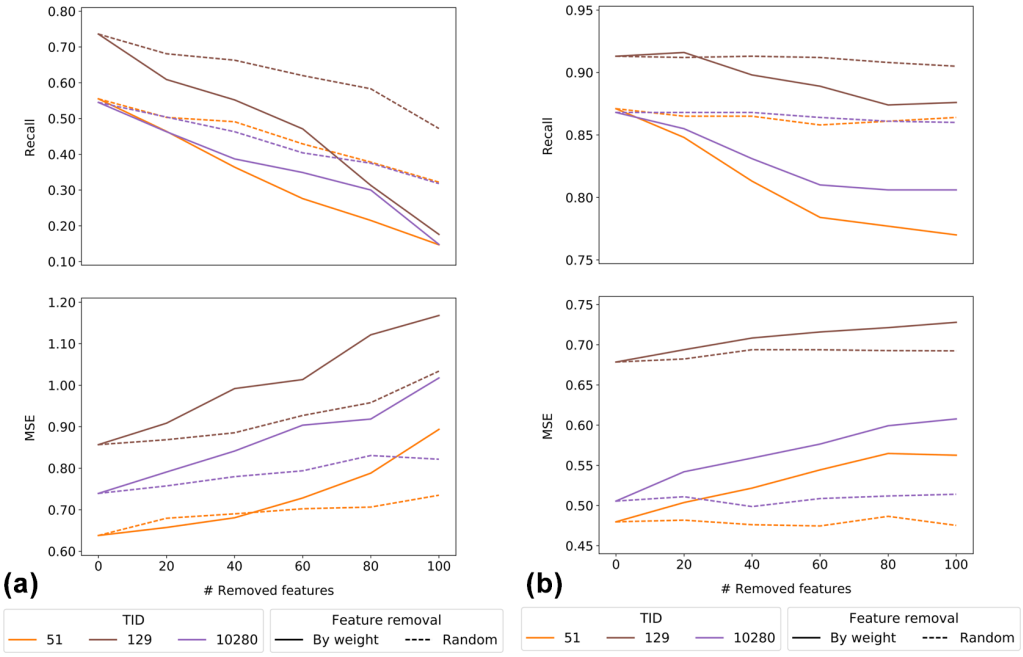


Figure 2. Effects of feature removal. For SVM and SVR, the effects of iterative fingerprint feature removal on recall of active compounds and MSE are reported for three exemplary activity classes (with TID values according to Table 1) and the (a) MACCS and (b) ECFP4 fingerprints. Features were randomly removed (dashed lines) or in the order of decreasing feature weights (solid lines).

were only relevant for either classification or regression. On average, 7 MACCS and 18 ECFP4 features were identified per activity class that had a high weight in at least 5 of the 10 SVM trials and a low weight in at least 5 SVR trials and vice versa. Among these, there were no MACCS and on an average one ECFP4 feature that exclusively had high/low weights in all SVM/SVR trials and vice versa. One possible explanation for such differences in feature relevance might be the composition of support vectors in SVM and SVR. Although SVM and SVR share a closely related methodological framework, support vectors for SVM and SVR are determined in different ways. To derive support vectors for regression, only active compounds are considered, whereas classification models are trained with active and inactive compounds, which also contribute to support vectors. Given these intrinsic differences, SVM and SVR models may prioritize different chemical descriptors for support vector compounds during the training stage.

In Figures 3b and 4b, exemplary MACCS and ECFP4 features are mapped onto the structures of compounds that were correctly predicted. In Figure 3b, MACCS features that were highly weighted in classification (blue color) or regression (red color) were mapped onto the same molecule, a thrombin inhibitor, illustrating that features critical for SVM or SVR are often mapped to different parts of the same substructure. In Figure 4b, ECFP4 features critical for classification (blue color) or regression (red color) are mapped to a serotonin 1A (5-HT1A) receptor agonist, showing that features important for classification (feature 638) or regression (201) are mapped to distant parts of this compound.

In principle, features relevant for SVM and SVR might be activity class-specific or shared by different classes. To identify features common to different classes, MACCS and ECFP4 features were determined that had a high weight in at least 5 of the 10 SVM or SVR trials per class. For SVM, on an average, 9 of such MACCS and 15 ECFP4 features were identified per activity class and for SVR, 14 MACCS and 35 ECFP4 features were identified. For SVM, a total of 38 MACCS and 47 ECFP4 highly weighted features were shared by two activity classes. For SVR, 56 MACCS and 116 ECFP4 features were shared by two classes. However, for SVM (SVR), only five (seven) MACCS and nine (three) ECFP4 features with at least five high weights were common to five or more activity classes. Thus, most features determining SVM and SVR predictions were weighted in a compound class-specific manner.

Furthermore, we also determined the number of features that were consistently highly weighted in all trials per activity class. For SVM, on an average, only two of such MACCS and five ECFP4 features were identified and for SVR, two and four MACCS and ECFP4 features, respectively, were identified. Thus, weights of most features with strong contributions to SVM and SVR predictions displayed some variations in different activity classes depending on the training sets.

2.4. Features with Different Signs. So far, only absolute feature weights were analyzed, which revealed many features that contributed differently to SVM and SVR. However, in SVM and SVR, feature weights may carry a positive or negative sign depending on how they influence the predictions. Features with a positive weight contribute to the prediction of active compounds in SVM and high potency values in SVR, whereas features with a negative weight contribute to the prediction of inactive compounds in classification and low potency values in regression. Thus, taking these signs into account further refines the view of differential feature contributions to SVM and SVR. Therefore, we also searched for features with high weights and different signs. Such features have opposite effects in SVM and SVR. Only few features were identified that had high weights in corresponding SVM and SVR trials but consistently different signs. Exemplary features with opposite effects in SVM and SVR are shown in Figure 5. For example, three MACCS features in Figure 5a contributed to the prediction of active

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

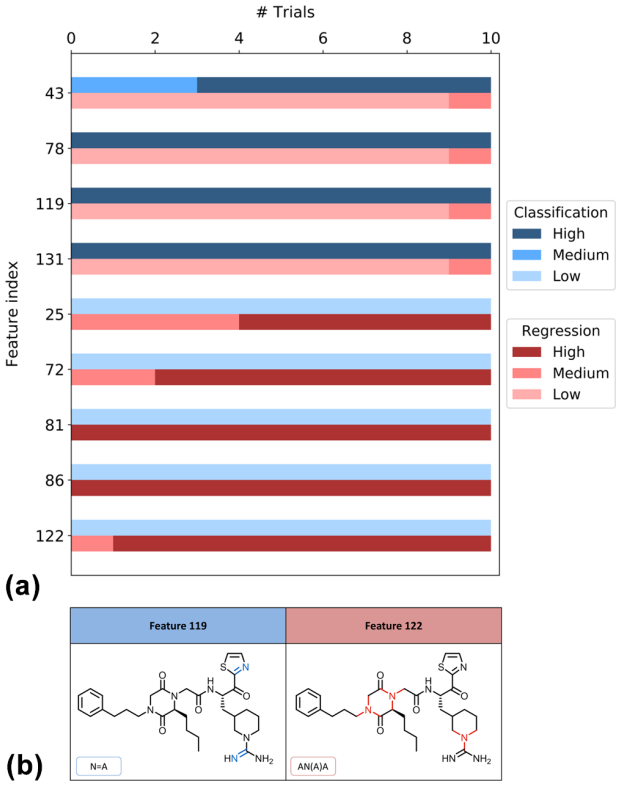


Figure 3. Distribution of MACCS feature weights and feature mapping. For an exemplary activity class (thrombin inhibitors, TID 11), (a) reports the distribution of weights of the selected features for SVM (classification, blue color) and SVR (regression, red color) over 10 trials. The color gradient represents the magnitude of feature weights (low, medium, or high). In (b), features that were highly weighted in SVM (blue color) and SVR (red color) are mapped on the same correctly predicted compound. In feature labels, "A" stands for any atom.

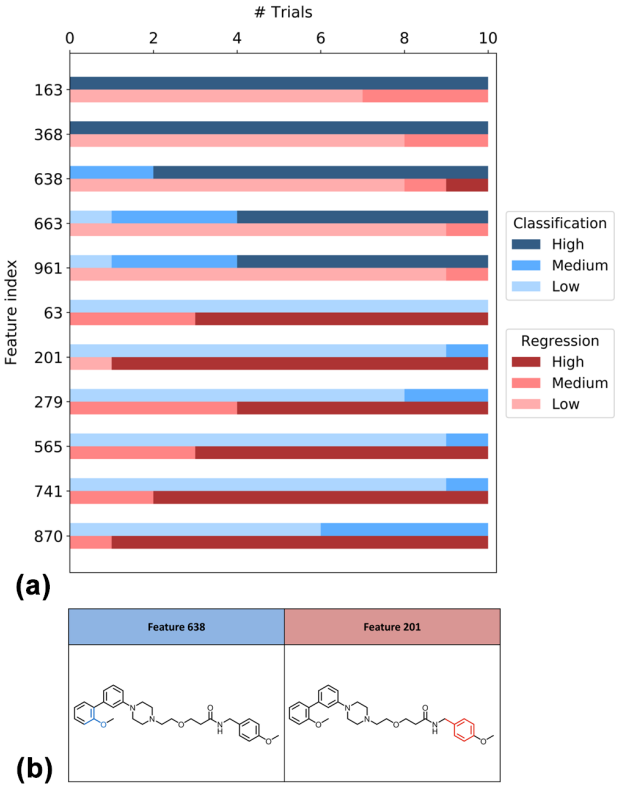


Figure 4. Distribution of ECFP4 feature weights and feature mapping. For an exemplary activity class (serotonin 1A (5-HT1A) receptor agonists, TID 51), (a) reports the distribution of weights of selected features for SVM (classification, blue color) and SVR (regression, red color) calculations over 10 trials. The color gradient represents the magnitude of feature weights (low, medium, or high). In (b), features that were highly weighted in SVM (blue color) and SVR (red color) are mapped on the same correctly predicted compound.

compounds but low potency values (dark green/light orange bars) and two to the prediction of inactive compounds but high potency values of active compounds (light green/dark orange bars). In Figure 5b, four ECFP4 features are shown that contributed to the prediction of active compounds and low potency values and one that contributed to the prediction of inactive compounds and high potency values. Among features with high weights in both SVM and SVR, as discussed above, sign inversion and opposite effects in SVM and SVR were exceptions.

2.5. Mapping of Highly Weighted Features. In Figure 6, highly weighted ECFP4 features are mapped on compounds from different activity classes that were correctly predicted using SVM and SVR. Atom environments were chosen for exemplary mapping because they have—by definition—a greater tendency to overlap than that involving discrete MACCS features. For an exemplary trial, features that had a high weight in the SVM and/or SVR model were mapped to the compounds shown. Figure 6a illustrates that only partly overlapping yet distinct atom environments led to the correct classification and potency value prediction of each compound.

The two thrombin inhibitors in Figure 6b are close structural analogues that are only distinguished by a heteroatom replacement in a ring and a fluorine substituent. As anticipated for highly similar compounds, these inhibitors shared a number of features that were highly weighted in classification and regression models. However, two features highly weighted for regression but not classification were mapped to the ring substructure distinguishing these compounds. Clearly, in contrast to the SVM model that assigned the same highly weighted features to both inhibitors, in accordance with their common activity, the SVR model accounted for the structural difference between these compounds. Hence, feature mapping also indicated that the fluorine substitution might be responsible for the higher potency of the inhibitor at the bottom, given its positive weight.

The two mu-opioid receptor ligands in Figure 6c are also analogous to each other but distinguished from each other by multiple substitutions at the upper and lower ring. In this case, few highly weighted features were present, only one of which was shared by the classification and regression models, covering the methyl substituent at the upper phenyl ring. Other highly weighted features in the models were distinct and mapped to different substructures. In the SVR model, a highly weighted feature with negative contribution matched a part of the upper phenyl ring including the methoxy substituent of the compound at the top, indicating that this substructure (but not the lower ring) was important for potency variation among analogues.

Taken together, these examples illustrate that comparative mapping of features highly weighted in SVM and SVR helps to

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

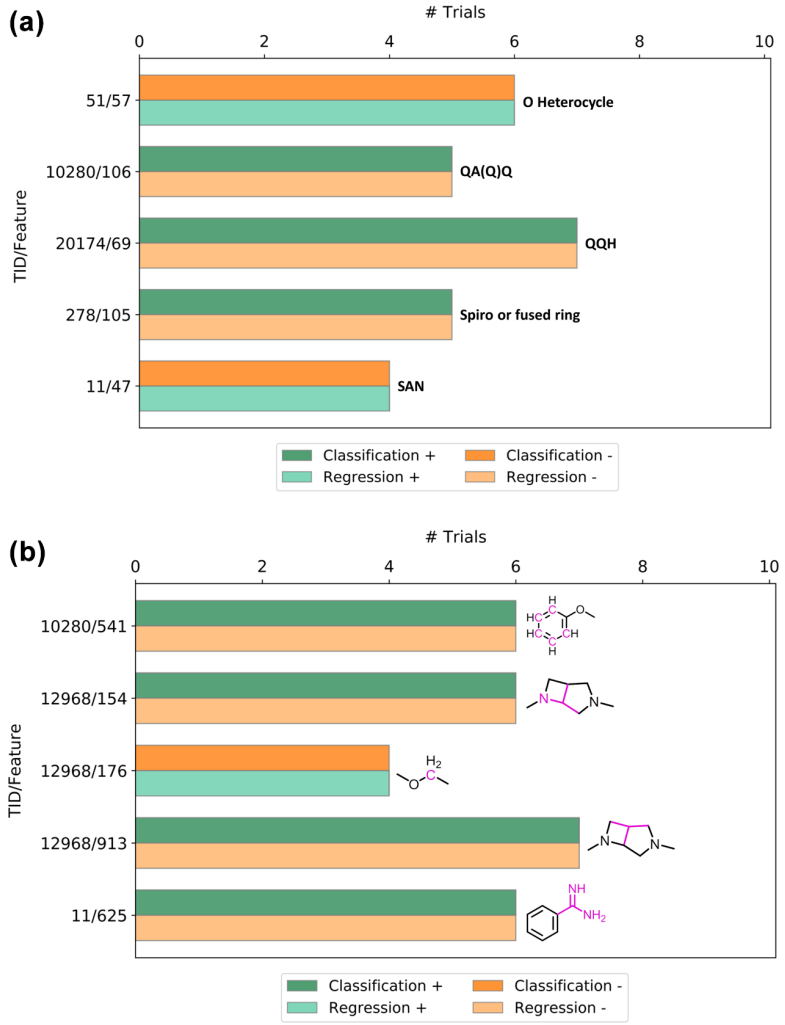


Figure 5. Highly weighted features with different signs. For selected activity classes and (a) MACCS and (b) ECFP4 features (TID/feature), the number of trials is reported in which the features had high weights but different signs (+, −) in SVM and SVR. Features with positive weights contribute to the correct prediction of active compounds (dark green color) or high potency values (light green color), whereas features with negative weights contribute to the prediction of inactive compounds (dark orange bars) or low potency values (light orange bars). Bars are labeled with MACCS features (A, any atom and Q, heteroatom) or mapped ECFP4 atom environments (pink color).

rationalize predictions made by classification and regression models and may reveal SAR information.

3. CONCLUSIONS

In this work, we have investigated and compared the relevance of different fingerprint features for the corresponding SVM and SVR models. The MACCS and ECFP4 fingerprints used herein capture the structural features of compounds in different ways. To these ends, feature weight analysis was carried out for well-performing classification and regression models over different compound classes. Because SVM and SVR share a common methodological framework, one might hypothesize that there should be considerable overlap between structural features that determine binary activity and potency value predictions. By contrast, systematic feature weight analysis revealed that features with high weights in SVM and SVR predominantly differed, a rather unexpected finding. In many instances, individual features contributed very differently to classification and regression, although features with strongly opposing effects were rare, as revealed by the analysis of positive and negative weights. SVM and SVR predictions are usually determined by feature combinations rather than individual features with high weights. Thus, features with medium weights also make contributions to predictions, albeit at a lesser magnitude than the most important ones. Therefore, as also demonstrated herein, mapping of highly weighted features is usually sufficient to identify molecular regions that are important for the activity-based classification and structural differences between compounds that are responsible for potency variation. Accordingly, mapping and comparing features that are highly weighted in SVM and SVR models help to better understand how individual features influence or determine predictions and thus alleviate the often-cited black box character of SVM, SVR, and other machine learning approaches that hinder model interpretation. Moreover, mapping of features that are highly weighted in SVR models onto compounds with correctly predicted potency values also points at SAR-informative regions in active compounds.

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

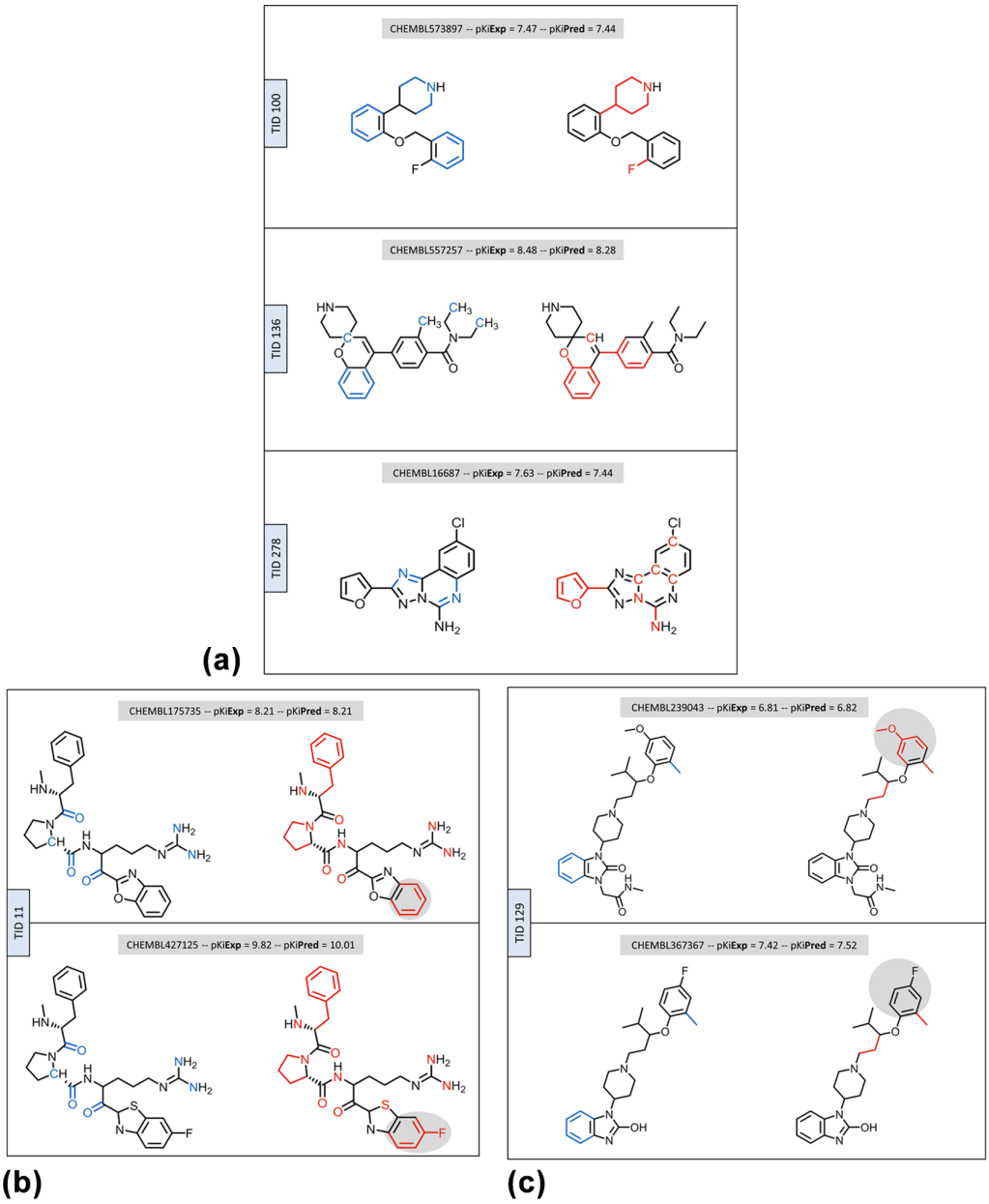


Figure 6. Mapping of highly weighted features. ECFP4 atom environments with high weights in classification and regression are mapped onto correctly classified compounds and potency prediction within 0.2 pK_i units. (a) shows individual compounds from three activity classes; (b,c) show pairs of analogues from two activity classes. Each compound is shown twice (side-by-side). On the left and right, features from classification (blue color) and regression (red color) are mapped, respectively. Single carbon atoms are displayed if they are a part of a mapped atom environment. In (b,c), substructures of analogues with feature differences are highlighted in gray color.

4. MATERIALS AND METHODS

4.1. Compound Data Sets. Different sets of compounds with activity against human targets were extracted from ChEMBL version 22.[17] Only compounds with numerically specified equilibrium constants (K_i values) for single human proteins with the highest assay confidence score were selected. If multiple K_i values for a compound and a target were available, they were averaged provided all values fell within the same order of magnitude; otherwise, the compound was discarded. Furthermore, compounds with a pK_i value below 5 were not selected to exclude borderline active compounds from modeling. In addition, this pK_i threshold also limited the range of potency values for SVR model building. Table 1 summarizes the 15 large activity classes that were selected. Each class contained at least 800 active compounds. In addition, for SVM modeling, 250 000 compounds were randomly selected from ZINC[18] as a pool of negative (inactive) training and test instances. From this pool, negative training and test sets were randomly sampled for all classification calculations.

4.2. Molecular Representation. Compounds were represented as MACCS[19] and ECFP4 fingerprints.[20] MACCS is a prototypic binary-keyed fingerprint comprising 166 bits, each of which accounts for the presence or absence of a structural fragment or pattern. ECFP4 is a representative

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

Table 1. Compound Data Sets[a]

TID	accession no.	target name	CPDs	median pK_i	IQR pK_i
11	P00734	thrombin	839	6.33	1.86
51	P08908	serotonin 1A (5-HT1A) receptor	1904	7.62	1.50
72	P14416	dopamine D2 receptor	2876	7.00	1.29
100	P23975	norepinephrine transporter	1099	6.82	1.60
129	P35372	mu-opioid receptor	2026	7.26	1.95
136	P41143	delta-opioid receptor	1547	7.11	1.97
137	P41145	kappa-opioid receptor	1930	7.28	2.07
138	P41146	nociceptin receptor	844	7.85	1.43
165	Q12809	HERG *Homo sapiens*	956	5.93	1.05
194	P00742	coagulation factor X	1476	8.05	2.80
278	P29275	adenosine A2b receptor	1187	7.23	1.43
10280	Q9Y5N1	histamine H3 receptor	2434	8.00	1.43
11362	P42336	PI3-kinase p110-α subunit	885	7.68	1.39
12968	O43614	orexin receptor 2	1040	6.70	1.57
20174	Q9Y5Y4	G protein-coupled receptor 44	833	7.65	1.90

[a]Composition of 15 compound activity classes is reported that were selected for SVM and SVR modeling. For each class, the ChEMBL target ID (TID), accession number, target name, and number of compounds (CPDs) are given. In addition, median and interquartile range (IQR) pK_i values are reported, which were calculated from the pK_i distribution of each activity class.

feature set fingerprint enumerating layered atom environments, which are encoded by integers using a hashing function. By design, ECFP4 has variable sizes, but it can be folded to obtain a fixed-length representation. For our calculations, ECFP4 was folded into a 1024-bit format using modulo mapping. Feature-to-bit mapping was recorded to enable mapping of fingerprint bits to compound structural features. Although modulo mapping assigns different features (atom environments) to identical bits, it is possible to trace environments and map them. Fingerprint representations were generated using in-house Python scripts based upon the OEChem toolkit.[21]

4.3. Support Vector Machine. For binary classification, training instances defined by a feature vector $x \in X$ and a class label $y \in \{-1,1\}$ are projected into the feature space X. For activity prediction, negative and positive examples represent inactive and active compounds for a given target, respectively. The SVM algorithm attempts to construct a hyperplane H such that the distance between the classes, the so-called margin, is maximized. This hyperplane is defined by a normal vector w and a scalar b using the expression $H = \{\mathbf{x}|\langle \mathbf{w},\mathbf{x}\rangle + b = 0\}$. For data that cannot be separated using a linear function, slack variables are added that permit training instances to fall within the margin or on the incorrect side of the hyperplane. To control the magnitude of allowed training errors, the cost or regularization hyperparameter C is introduced to balance margin size and classification errors. This represents a primal optimization problem that can be expressed in a dual form using Lagrange multipliers α_i (Lagrangian dual problem). Its solution yields the normal vector of the hyperplane $w = \sum_i \alpha_i y_i \mathbf{x}_i$. Training examples with nonzero coefficients represent the support vectors and correspond to data points of one class that are closest to the other, that is, those that lie on the margin of the hyperplane. Once the hyperplane is derived, test data are projected into the feature space and classified according to the side of the plane on which they fall, that is, $f(\mathbf{x}) = \mathrm{sgn}(\sum_i \alpha_i y_i \langle \mathbf{x}_i,\mathbf{x}\rangle + b)$, or ranked using the real value, that is, $g(\mathbf{x}) = \sum_i \alpha_i y_i \langle \mathbf{x}_i,\mathbf{x}\rangle + b$.[9]

4.4. Support Vector Regression. Training samples for SVR are defined by a feature vector $x \in X$ and a numerical label $y \in \mathbf{R}$.[10,11] If SVR is applied to potency prediction, the numerical label is the pK_i value of the compound. SVR maps the training data as close as possible to the quantitative output y by deriving a regression function of the type $f(\mathbf{x}) = \langle \mathbf{w},\mathbf{x}\rangle + b$. Tolerated deviations from the observed and predicted values of training data are at most ε, and larger errors are penalized. In SVR, the relaxation of error minimization problem is also controlled by a hyperparameter C, which penalizes large slack variables or deviations from the so-called ε tube. By solving the optimization problem with a Lagrange reformulation, the normal vector is derived and the prediction function is expressed as $f(\mathbf{x}) = \sum_i \alpha_i \langle \mathbf{x}_i,\mathbf{x}\rangle + b$.

4.5. Kernel Function. When accurate data separation is not feasible in the X space, the standard scalar product $\langle \cdot,\cdot \rangle$ is replaced by a kernel function $K(\cdot,\cdot)$. Conceptually, the kernel function represents the scalar product in a high-dimensional space W in which the data might become linearly separable, without the need to compute an explicit mapping to W. This approach is known as the "kernel trick"[13] that is applied in both SVM and SVR. In chemoinformatics, one of the most popular kernels for fingerprint representations is the Tanimoto kernel[22] that was also used herein

$$K(\mathbf{u}, \mathbf{v}) = \frac{\langle \mathbf{u}, \mathbf{v}\rangle}{\langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle - \langle \mathbf{u}, \mathbf{v}\rangle}$$

4.6. Feature Weight Analysis. In the SVM model, different weights are assigned to molecular descriptors (features), which correspond to the coefficients of the primal optimization problem. The linear kernel (scalar product) allows direct determination of feature weights from the dual problem coefficients and support vectors. By contrast, direct access to feature weights is not possible when using nonlinear kernel functions because an explicit mapping into the high-dimensional feature space is not computed. However, for the Tanimoto kernel, feature weight analysis can be adapted from the linear case according to which the importance of a feature depends on the coefficients of those support vectors that contains the feature.[16] To account for the nonlinearity of the Tanimoto formalism, a normalization factor is included for each individual support vector by dividing the feature weight contribution by the total number of features present in each support vector

$$\mathrm{FW}(d) = \sum_{i=1}^{m} \frac{\alpha_i v_{id}}{\sum_{d^*=1}^{D} v_{id^*}}$$

Here, $\mathrm{FW}(d)$ is the feature weight for feature d, D is the dimensionality, m is the number of support vectors, and v_i and α_i are the support vector coefficients of the dual problem solution.

Feature contributions are not constant across feature space and depend on the fingerprint that is used.[16] However, adaptation of feature weight analysis from the linear case with normalization yields an average weight, indicating the importance of each feature. Highly weighted fingerprint features can then be mapped to compound structures.[16]

4.7. Calculations and Data Analysis. Each activity class was randomly divided into training and test (prediction) sets comprising 700 and 100 compounds, respectively, following

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

previously derived guidelines for relative training and test set composition.[23] For SVM, 700 and 100 compounds from ZINC database were randomly selected as negative training and test instances, respectively. For SVR, the same positive training data were used in each case (but no negative data). For each activity class and SVM/SVR calculation protocol, 10 independent trials were carried out, and the results were averaged.

For SVM and SVR models, the hyperparameter C was optimized using 10-fold cross-validation on training data using candidate values of 0.01, 0.1, 1, 5, 10, 20, 50, and 100. For SVM, hyperparameter optimization was guided by maximizing the F1 score; for SVR, optimization aimed to minimize the MAE.

$$\mathrm{F1} = \frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Here, n is the number of samples (see also MSE given below).

Following hyperparameter optimization, feature weight analysis was carried out for classification and regression models. Weights were categorized as *high*, *medium*, or *low*, depending on whether their absolute value was at least 50, 25−50%, or less than 25% of the maximum weight observed for a given SVM model, respectively.

Binary activity (active/inactive) and potency values of test compounds were predicted, and model performance was estimated using different figures of merit. For SVM, the F1 score, AUC, and the recall of active compounds among the top 1% of the ranked test set were determined. For SVR, MAE, MSE, and the Pearson correlation coefficient between the observed and predicted pK_i values were calculated.

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Calculation and data analysis protocols were implemented in Python using *Scikit*-learn.[24]

AUTHOR INFORMATION

Corresponding Author

*E-mail: bajorath@bit.uni-bonn.de. Phone: 49-228-2699-306 (J.B.).

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Author Contributions

The study was carried out and the manuscript was written with contributions from all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.-P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 676434, "Big Data in Chemistry" ("BIGCHEM", http://bigchem.eu). The article reflects only the authors' view, and neither the European Commission nor the Research Executive Agency (REA) is responsible for any use that may be made of the information it contains. We thank the OpenEye Scientific Software, Inc., for providing a free academic license of the OpenEye toolkit.

REFERENCES

(1) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Chemoinformatics: Quo Vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413−1437.

(2) Vogt, M.; Bajorath, J. Chemoinformatics: A View of the Field and Current Trends in Method Development. *Bioorg. Med. Chem.* **2012**, *20*, 5317−5323.

(3) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Comput. Chem.* **2001**, *26*, 5−14.

(4) Geppert, H.; Vogt, M.; Bajorath, J. Current Trends in Ligand-Based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation. *J. Chem. Inf. Model.* **2010**, *50*, 205−216.

(5) Heikamp, K.; Bajorath, J. Support Vector Machines for Drug Discovery. *Expert Opin. Drug Discovery* **2014**, *9*, 93−104.

(6) Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273−297.

(7) Burges, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121−167.

(8) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.

(9) Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases Using a Support Vector Machine. *J. Chem. Inf. Model.* **2005**, *45*, 549−561.

(10) Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; Vapnik, V. Support Vector Regression Machines. *Adv. Neural Inform. Process. Syst* **1997**, *9*, 155−161.

(11) Smola, A. J.; Schölkopf, B. A Tutorial on Support Vector Regression. *Stat. Comput.* **2004**, *14*, 199−222.

(12) Balfer, J.; Bajorath, J. Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis. *PLoS One* **2015**, *10*, No. e0119301.

(13) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*; Pittsburgh, Pennsylvania, 1992; ACM: New York, 1992; pp 144−152.

(14) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(15) Hansen, K.; Baehrens, D.; Schroeter, T.; Rupp, M.; Müller, K.-R. Visual Interpretation of Kernel-Based Prediction Models. *Mol. Inf.* **2011**, *30*, 817−826.

(16) Balfer, J.; Bajorath, J. Visualization and Interpretation of Support Vector Machine Activity Predictions. *J. Chem. Inf. Model.* **2015**, *55*, 1136−1147.

(17) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.

(18) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.

(19) *MACCS Structural Keys*; Accelrys: San Diego, CA, 2011.

(20) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(21) *OEChem TK*, version 2.0.0; OpenEye Scientific Software: Santa Fe, NM, 2015.

(22) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Network.* **2005**, *18*, 1093−1110.

(23) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.
(24) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

DOI: 10.1021/acsomega.7b01079
ACS Omega 2017, 2, 6371−6379

Summary

SVM and SVR modeling was performed for distinct compound activity classes and fingerprint features determining performance were identified to enable interpretation. A systematic analysis reflected considerable differences between highly weighted features in classification and regression models. Interestingly, predictions were generally determined by combinations of features rather than single descriptors, but the mapping of highly weighted features was often sufficient to identify activity-relevant regions on the compound 2D structure. Structural patterns prioritized in binary activity and potency predictions were mapped onto compounds and SAR-informative regions were examined. Important molecular regions of compounds were usually different for SVM and SVR model predictions. Hence, despite sharing the same algorithmic basis, SVM and SVR use distinct structural patterns from compounds to predict activity and potency, respectively.

In the following chapter, a new methodology to interpret model predictions is introduced. Rather than focusing on a single ML method, the proposed and analyzed approach is applicable to any ML model.

Chapter 7

Interpretation of Compound Activity Predictions from Complex Machine Learning Models Using Local Approximations and Shapley Values

Introduction

In the previous chapter, a step towards a better interpretability of SVM and SVR models was presented and feature weighting was systematically explored. Other ML methods might be preferred under certain circumstances. Thus, access to the feature prioritization of distinct ML models is crucial. However, many ML methods, especially DNNs, are complex and have a black box character, which hinders interpretability. The rationalization of bioactivity predictions has an additional layer of complexity compared to other fields, which is compound representation. Even though feature importance is extracted, it has to be presented to the user in an intuitive and understandable way. In this chapter, a new method based on local approximations and the Shapley values concept from game theory is proposed to interpret activity predictions from any ML model. Exemplary algorithms and visualizations are reported to establish proof-of-principle.

Journal of Medicinal Chemistry

Article

Cite This: J. Med. Chem. XXXX, XXX, XXX−XXX

pubs.acs.org/jmc

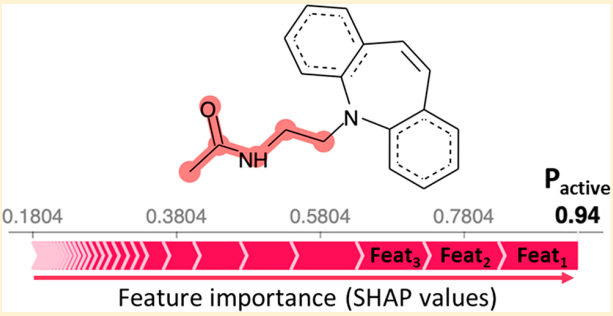
Interpretation of Compound Activity Predictions from Complex Machine Learning Models Using Local Approximations and Shapley Values

Raquel Rodríguez-Pérez[†,‡] and Jürgen Bajorath*[,†]

[†]Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Endenicher Allee 19c, D-53115 Bonn, Germany

[‡]Department of Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, Birkendorfer Straße 65, 88397 Biberach an der Riß, Germany

ABSTRACT: In qualitative or quantitative studies of structure−activity relationships (SARs), machine learning (ML) models are trained to recognize structural patterns that differentiate between active and inactive compounds. Understanding model decisions is challenging but of critical importance to guide compound design. Moreover, the interpretation of ML results provides an additional level of model validation based on expert knowledge. A number of complex ML approaches, especially deep learning (DL) architectures, have distinctive black-box character. Herein, a locally interpretable explanatory method termed Shapley additive explanations (SHAP) is introduced for rationalizing activity predictions of any ML algorithm, regardless of its complexity. Models resulting from random forest (RF), nonlinear support vector machine (SVM), and deep neural network (DNN) learning are interpreted, and structural patterns determining the predicted probability of activity are identified and mapped onto test compounds. The results indicate that SHAP has high potential for rationalizing predictions of complex ML models.



INTRODUCTION

Compound bioactivity prediction and structure−activity relationship (SAR) analysis are major applications of machine learning (ML) in pharmaceutical research.[1−6] Supervised ML methods are trained to search for structural patterns that differentiate between active and inactive compounds. Since prospective predictions using such activity models provide decision support and guidance for compound exploration and design, there is a high level of interest in obtaining accurate models and in rationalizing their predictions.[7−9] However, while much attention has been paid to improving the predictive performance of ML models, interpreting the predictions currently is an underinvestigated area, despite its high relevance.[10,11]

While statistical performance measures and method validation procedures are of critical importance for ML, they do not provide scientific insights into predictions, which can typically only be achieved on the basis of expert knowledge. On the other hand, rationalizing model decisions would assign priority to meaningful predictions, help to extract knowledge from ML models, and also increase the acceptance of and confidence in predictions in pharmaceutical research.[5,12,13] In activity prediction, model interpretation generally relies on the identification of chemical features that determine predictions.[14,15] For simplistic linear (Q)SAR models, the interpretation of structural and/or property changes that modulate activity is often straightforward.[13] However, the situation fundamentally changes when ML models become complex, which often increases predictive performance at the expense of interpretability, ultimately leading to the frequently quoted "black-box" character of ML model and their predictions.[13,15] For example, the random forest (RF)[16] and support vector machine (SVM)[17] algorithms are robust and well-performing ML methods that have become very popular in the field. However, RF and SVM models are very difficult to interpret and exhibit black-box character, for different reasons. In the case of RF, this is largely due to the generation of large decision tree ensembles, leading to statistically driven decisions; in the case of SVM, black-box character results from the use of nonlinear kernels to facilitate data mapping into feature reference spaces of increasing dimensionality.[18]

Currently, compound activity data grow at unprecedented rates,[19,20] leading to emerging big data phenomena in medicinal chemistry[19] and catalyzing the application of deep learning (DL)[21] strategies for activity prediction. Among ML methods, DL architectures have shown particular promise in data-rich fields such as image analysis[22] or natural language

Special Issue: Artificial Intelligence in Drug Discovery

Received: July 8, 2019
Published: September 12, 2019

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

processing[23] and deep neural networks (DNNs) also gain increasing popularity in chemical informatics and drug design.[24−26] Although some successful applications in compound design and activity prediction using DNNs have been reported, it remains unclear at present whether DL might provide a consistent advantage over other ML methods in at least some application scenarios.[27−31] However, DNNs have higher complexity than other ML models and their black-box character is notorious. Any form of model diagnostics becomes essentially prohibitive for DNNs, and domain experts struggle to understand why DNN models succeed or fail,[32] which hinders advances in the field.

Several interpretation strategies have been proposed to reduce the black-box nature of ML models.[13] These approaches can essentially be divided into model-specific and model-agnostic (or model-independent) strategies. As a model-specific approach, feature weighting has been applied to better understand predictions of SVM[18,33] and RF models.[34] As a model-agnostic method, sensitivity analysis can be used to investigate the influence of systematic feature value changes on the model output.[35] Sensitivity analysis has been applied to different ML algorithms including neural networks[36] but becomes quickly inefficient with increasing dimensionality of models and has thus hardly been used in chemical informatics.[13] An exception is provided by investigating partial derivatives as a form of local sensitivity analysis that has been applied in QSAR modeling.[13] Here, for a given compound, a perturbation is introduced to an individual feature and calculation of the partial derivative provides an estimation of its contribution to model performance.[37,38] However, effective use of partial derivatives is also limited given its intrinsic focus on individual features. A principal advantage of model-agnostic over model-specific interpretation approaches, if they can be established, is that model-agnostic analysis alleviates the need to balance model performance and interpretability.[39,40]

In this work, we introduce a conceptual new agnostic interpretation method for ML models of arbitrary complexity used for activity prediction. The Shapley additive explanations (SHAP) approach[41] is an extension of local interpretable model-agnostic explanations (LIME)[42] according to which feature weights are represented as Shapley values from game theory.[43] As shown herein, SHAP is capable of interpreting activity predictions from complex ML models. Features that increase or reduce the probability of predicted activity are identified and mapped onto molecular graphs to identify and visualize structural patterns that determine predictions.

RESULTS

Principles of Explanation Models and the LIME Approach. *Explanation Model.* The principal goal of an explanation model g is to simplify or locally approximate a complex model f that cannot be directly interpreted. Additive feature attribution methods generate an explanation model via a linear function of binary variables, as shown in eq 1:

$$g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{1}$$

where $x' \in \{0,1\}^M$, M is the number of input features, and $\phi_i \in \mathbb{R}$.[42] The presence or absence of a feature value impacts the model, which can be referred to as a feature contribution (ϕ_i). Accordingly, a weight must be assigned to each variable. Therefore, the SHAP method has been devised, which represents an extension of the LIME approach.

LIME. The LIME methodology generates the explanation ξ of an instance x according to eq 2:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}}\, \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2}$$

where G is a class of interpretable (linear) models, $\mathcal{L}$ is the loss function to minimize, π_x the proximity measure between an instance z and x (kernel defining locality), and $\Omega(g)$ an optional regularization term to control (limit) model complexity.[42]

For the interpretation of a given test instance x, the following procedure is applied.

(i) Artificial samples are obtained by permuting features of the test instance x.
(ii) These samples are weighted by the value of a kernel calculated for them and x.
(iii) A model g is trained to predict $f(x)$ with coefficients corresponding to feature importance estimates.

It follows that LIME builds a linear model g in a feature region proximal to the test instance, although model f might be nonlinear, as illustrated in Figure 1. This figure also shows that samples similar to x receive high weights, due to the application of the kernel function. This conceptual framework provides the basis for the development of the SHAP methodology detailed in the following.

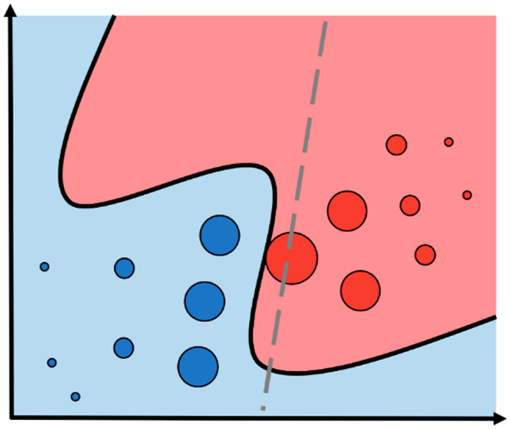


Figure 1. Local approximations for model interpretation. The active (red) and inactive (blue) regions in feature space correspond to the decision function of the complex model f. The dashed gray line represents the decision function of the simple explanation model g, which locally approximates the global model. The largest red dot is the active instance x to be explained, while the other dots are artificial samples that are weighted by the kernel function with respect to x.

SHAP Method. *Shapley Value Concept.* Shapley values from cooperative game theory provide a connection between LIME and the SHAP methodology. Specifically, Shapley values were introduced in the 1950s to measure contributions of individual players to a collaborative game.[43] They provide a theoretically grounded partition of payoff or credit among members of a team by considering the average of all contributions made by a player.[43] This concept can be applied to feature attributions by considering the success of a team (or total credit) as an output (prediction), and each player's contribution (or player's payoff) as the feature importance.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

Therefore, in this context, Shapley values facilitate the distribution of a model's prediction resulting from an input feature vector over the individual features.

To obtain the contribution of a feature i, all operations by which a feature might have been added to the set ($N!$) and a summation over all possible sets (S) is considered. For any feature sequence, the marginal contribution through addition of feature i is given by $[f(S\cup\{i\}) - f(S)]$. The resulting quantity is weighted by the different possibilities the set could have been formed prior to feature i's addition ($|S|!$) and the remaining features could have been added ($(|N| - |S| - 1)!$). Hence, the importance of a given feature i is defined by eq 3:

$$\phi_i = \frac{1}{N!} \sum_{S\subseteq N\setminus\{i\}} |S|!(|N| - |S| - 1)![f(S \cup \{i\}) - f(S)] \quad (3)$$

It follows that Shapley values represent a unique way to divide a model's output among feature contributions satisfying three axioms: *local accuracy* (or additivity), *consistency* (or symmetry), and *nonexistence* (or null effect).

SHAP Formalism. Additive feature attribution methods typically do not consider two properties that are of high relevance for assessing feature importance, i.e., *local accuracy* and *consistency*, as referred to above. Taking these axiomatic properties into account was a main motivation for proposing the SHAP concept.[41] The property *local accuracy* forces the sum of individual feature attributions to be equal to the original model prediction. In addition, *consistency* ensures that feature importance correctly accounts for different models on a relative scale. Hence, if a change in a feature value has larger impact on a model A than a model B, feature importance should be larger in A. These properties can be considered by expressing feature weights as Shapley values.[43]

A weighting procedure for artificial samples is a key aspect for connecting Shapley values to the LIME approach, which allows the approximation of Shapley values. In LIME, heuristic choices are made to select $\mathcal{L}$, $\Omega(g)$, and π_x. By contrast, the SHAP method introduces a special kernel function that is related to the Shapley value definition, assuming that feature weights follow the two axioms of interpretability.[41] Specifically, SHAP uses the following procedure for interpreting an instance x:

(i) Training data is organized by k-means clustering and the k samples are weighted by the number of training instances they represent. These samples constitute a background data set with "typical" feature values.
(ii) Artificial samples are obtained by replacing features of the test instance x with the values from the background data set.
(iii) These artificial samples are weighted by the value of the SHAP kernel calculated for them and x.
(iv) A weighted linear regression model g is trained to predict $f(x)$. The model coefficients are Shapley values corresponding to feature importance estimates.

Sampling all possible feature subsets is time-consuming. Therefore, the input vector is permutated for an individual prediction by setting its features on and off, thereby examining feature influence. Herein, 1000 artificial samples were generated in each case and missing features were simulated by replacing them with the values obtained from a k-means clustering of the training set (k = 100). A feature obtained a large weight if its replacement with an artificial (non-informative) value led to a significant change in model output. Weights of artificial samples were determined according to the number of feature-addition sequences that a given subset accounted for on the basis of the SHAP kernel. Local linear regression resulted in coefficients representing feature weights as Shapley values. These weights indicate how important a feature is for a given prediction and include the direction (sign) of feature influence. The expected explanatory value is calculated as the mean of the model output probability over training set instances. For a given compound, the original output probability (of activity) given by model f is then retrieved by summing the expected (or base) value and all SHAP values.

Model Building and Analysis Strategy. ML models were built for 10 activity classes summarize in Table 1. These classes were assembled on the basis of specific structural and activity data selection criteria detailed in the Experimental Section. As negative training and test instances, compounds with unknown activity status were considered inactive and randomly assembled, as also reported in the Experimental Section. Feature contributions were systematically calculated for test set compounds. First, model performance for three different ML algorithms and two molecular representations is reported. Then, the effect of feature removal is investigated. SHAP results for RF models are compared to Gini importance, and the relationship between SHAP values obtained for different ML methods is examined. Next, representative examples are shown to illustrate SHAP results. Individual predictions using ML algorithms are interpreted and differences in feature importance are explored. Furthermore, for individual predictions, important (fingerprint) features are mapped onto compounds and visualized.

Table 1. Compound Data Sets[a]

CHEMBL identifier	target	no. compounds	no. ASs	mean pK_i
229	α-1a adrenergic receptor	243	80	7.8
4860	Apoptosis regulator Bcl-2	283	67	9.0
244	Coagulation factor X	679	154	7.5
264	Histamine H3 receptor	955	216	8.0
237	κ opioid receptor	716	160	7.5
344	Melanin-concentrating hormone receptor 1	409	73	7.4
259	Melanocortin receptor 4	443	57	6.9
1946	Melatonin receptor 1B	285	70	8.2
233	μ opioid receptor	831	194	7.6
4792	Orexin receptor 2	399	81	6.9

[a]Reported are the CHEMBL identifier, target name, number of compounds, number of analog series (ASs), and mean pK_i values for 10 compound activity classes.

While our study is focused on method development and evaluation, it is essential to carry out the analysis on newly generated ML models and their predictions to ensure independence of ML assessment (rather than reliance on previously reported models) and reproducibility of the results.

Global Model Performance. Accurate predictions are a key requirement for meaningful model interpretation. If ML models are not predictive, the prioritized chemical patterns do not correlate well with activity prediction. Thus, initially, the predictive performance of SVM, RF, and DNN models over different compound activity classes was determined. Models were built on the basis of the state-of-art ECFP4 and easy-to-

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

Table 2. Classification Performance[a]

	ECFP4			MACCS		
metric	SVM	RF	DNN	SVM	RF	DNN
AUC	0.98 (0.02)	0.98 (0.02)	0.98 (0.02)	0.97 (0.02)	0.97 (0.02)	0.97 (0.02)
BA	0.89 (0.30)	0.84 (0.05)	0.91 (0.03)	0.88 (0.04)	0.84 (0.04)	0.89 (0.03)
MCC	0.87 (0.03)	0.80 (0.07)	0.88 (0.03)	0.83 (0.06)	0.79 (0.06)	0.81 (0.05)

[a]Area under the ROC curve (AUC), balanced accuracy (BA), and Matthew's correlation coefficient (MCC). Mean (and standard deviation) values are reported across 10 activity classes. Performance values are given for two molecular representations (ECFP4 and MACCS) and three ML methods (SVM, RF, and DNN).

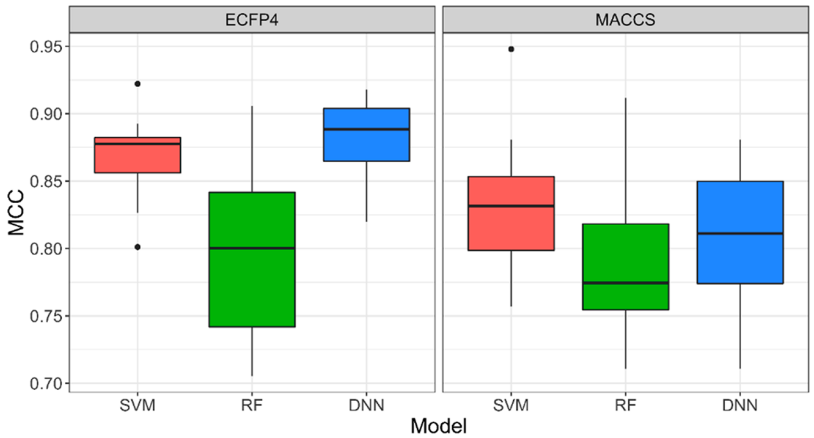


Figure 2. Global classification performance. Boxplots show value distributions of Matthew's correlation coefficient (MCC) across 10 compound data sets using SVM (red), RF (green), DNN (blue) models and two fingerprints (ECFP4, left; MACCS, right).

understand MACCS fingerprints. Table 2 reports average model performance on the basis of the AUC, BA, and MCC measures. Overall, activity predictions for the 10 activity classes were consistently accurate for the investigated methods and molecular representations, hence providing a sound basis for model analysis. Overall, rankings of test compounds yielded AUC values greater than 0.9, BA of ∼90%, and MCC values of around 0.8 or larger. Figure 2 reports the distribution of MCC values for all ML method/representation combinations. As anticipated, MCC values were larger for ECFP4 than MACCS, albeit by a confined margin. In addition, RF predictions were generally slightly less accurate than SVM and DNN predictions. Although hyperparameter combinations were optimized (see Experimental Section), alternative parameter settings did not have a large influence on the predictions because active compounds were overall easily distinguishable from random ZINC examples. Taken together, the results showed that the test system setup was appropriate for our proof-of-principle investigation of a new model interpretation methodology.

Feature Importance. To interpret the prediction for a test compound, SHAP calculations were carried out resulting in a set of feature weights. Initially, the distributions of ECFP4 features with nonzero SHAP values (feature weights) were determined for all test compounds. Figure 3a shows how many feature variables were contributing to the RF, SVM, and DNN predictions of individual test instances. SVM and DNN distributions were centered on smaller values than RF, indicating that more features were required to provide local explanations for RF predictions. The average number of features with nonzero SHAP values for a test instance was 68 and 67 for SVM and DNN, respectively, and 96 for RF. These numbers represented less than 10% of the entire ECFP feature population obtained for the activity classes, revealing that limited numbers of features were important for the predictions.

Because some features with nonzero SHAP values might not contribute significantly to predictions, absolute SHAP values of features were normalized with respect to the total sum of SHAP values for a given prediction, resulting in a percentage value for a feature. This percentage represents the fraction of feature weights that a given variable is accounting for, considering both negative and positive contributions. Thus, the cumulative SHAP percentage for a given number of top-ranked features can be calculated per test instance. Figure 3b shows the distributions of cumulative SHAP percentage values for different numbers of top-ranked features. The distributions were nearly identical for all three ML methods and showed that the top-1, -5, -10, and -20 ranked features generally corresponded to 7%, 25%, 40%, and 60% of the cumulative (total) feature weights of a prediction, respectively. These findings indicated that top-ranked features provided sufficient information for model interpretation.

Feature Elimination. The next step was exploring whether SHAP values indeed identified features that were important for predictive performance. Therefore, for each data set and ML model, SHAP values were calculated for all test compounds. Then, absolute SHAP values were averaged over test compounds to obtain an ECFP4 feature importance ranking. Finally, features were systematically eliminated, either randomly or in the order of SHAP ranking, and the ML models were generated again using the reduced feature sets. Following this protocol, RF, SVM, and DNN control models were built after removal of 4, 10, 80, 160, and 320 ECFP4 features. Figure 4 shows the median MCC values of SVM, RF, and DNN models across all activity classes for varying numbers of features. The results revealed that random elimination of up

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

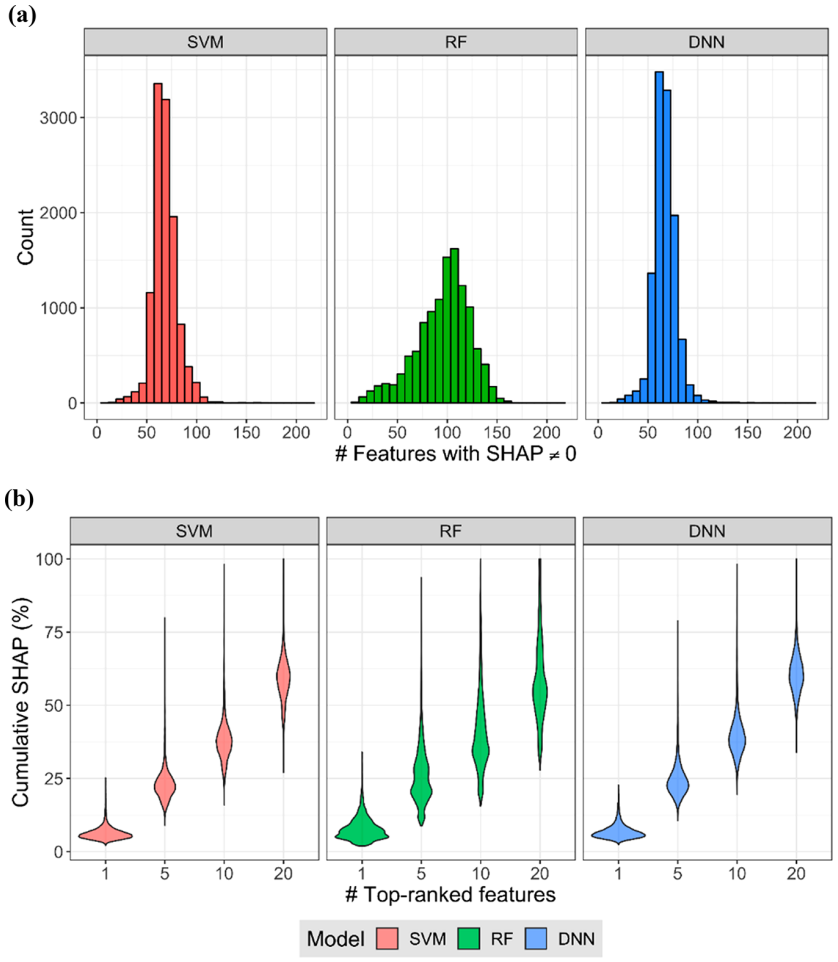


Figure 3. Distribution of SHAP values. (a) shows distributions of features with nonzero SHAP values over all test compounds (Count) for RF, SVM, and DNN predictions. (b) shows distributions of cumulative SHAP percentage values for different numbers of top-ranked features.

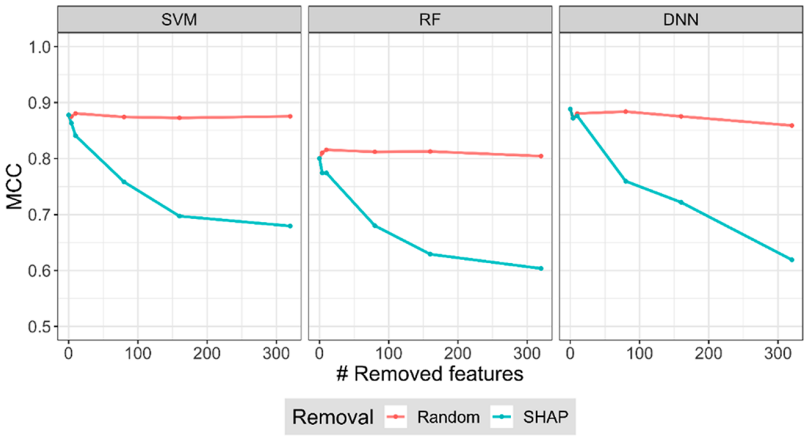


Figure 4. Feature removal. MCC values are shown for varying numbers of ECFP4 features, which were removed randomly (red) or according to decreasing mean absolute SHAP values (blue). Results are shown for SVM (left), RF (center), and DNN (right) models.

to 320 features did not notably affect the performance of ML models, which remained essentially constant, providing further evidence for general ECFP4 feature redundancy. By contrast, removal of features with large average SHAP values led to a substantial decrease in model performance for the three ML algorithms.

For all ML methods, the MCC value distribution after feature removal according to SHAP values was significantly larger than the one after random elimination (Wilcoxon test, p-values $\ll 0.0001$). These results confirmed that SHAP values provided a quantitative measure of feature importance for predictions using different ML models.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

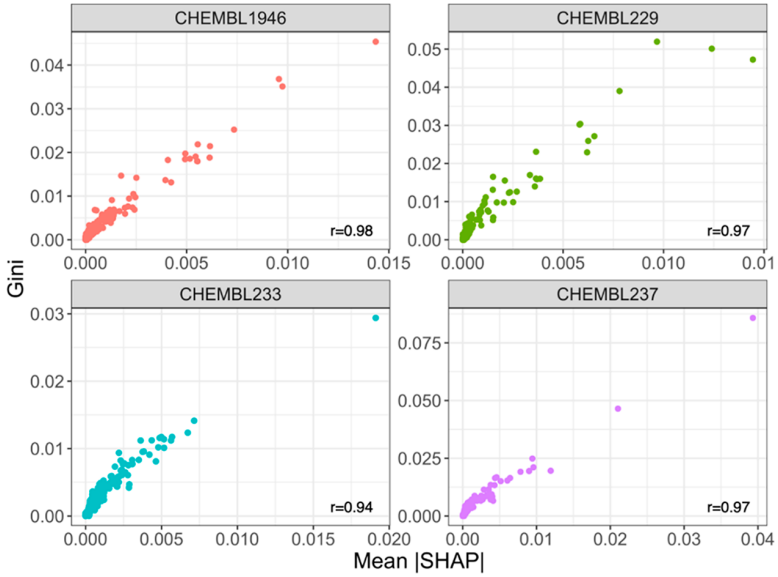


Figure 5. Relationship between SHAP and Gini importance. For four activity classes, RF models were built to predict the activity of test compounds. For each ECFP4 feature, mean absolute SHAP values for test compounds and Gini importance are reported for RF models. In addition, the correlation coefficient for feature weighting methods is reported.

SHAP versus Gini Importance. In an additional control calculation, SHAP feature weights were compared to Gini importance,[34] which has become a popular measure for the assessment of variable importance in decision tree-based methods such as RF. Gini importance is equivalent to the mean decrease in Gini "impurity", which measures the probability of a new sample to be incorrectly classified at a given node in a tree weighted by the proportion of samples representing the data partition. Gini feature importance values were calculated during RF model building[56] and were thus not dependent on test instances. Gini calculations yielded absolute (nonsigned) values, which were thus compared to mean absolute SHAP values determined from predictions of all test compounds. Figure 5 compares feature weights obtained using both approaches for RF models of four activity classes. Each point represents the weights for a given feature using SHAP and Gini importance. There was strong correlation between these orthogonal feature weights (i.e., one derived on the basis of training, the other on the basis of testing), without any outlier or notable inconsistency. However, while Gini feature importance is confined to decision tree methods, SHAP is generally applicable.

SHAP Comparison. Next, relationships between SHAP values for the same compound sets and different ML methods were examined. Despite algorithmic differences, which might affect variable prioritization, ML models with predictive power should detect similar chemical patterns that differentiate between active and inactive compounds for a given molecular representation.

Figure 6 shows mean absolute SHAP values for test compounds from two activity classes. SHAP values originating from SVM and RF models were compared in a pairwise manner to corresponding values from DNN models based upon the ECFP4 (Figure 6a) and MACCS (Figure 6b) representations. Correlation coefficients were high, ranging from 0.90 to 0.98 for ECFP4 and from 0.83 to 0.95 for MACCS. Highly weighted features were consistently prioritized for models generated with all ML methods, thus confirming algorithm-independent consistency of feature relevance. We note that features that are important in a local explanation model might not be globally relevant. Therefore, some features influencing individual predictions might yield small (but nonzero) mean SHAP values because they were not prioritized in the majority of explanation models.

Feature weight relationships between different ML methods were also examined across all activity classes. Therefore, correlation between mean absolute SHAP values for models generated with different methods was determined. The resulting distributions or correlation values are shown in Figure 7. All method combinations displayed high correlation of feature importance, especially SVM and DNN, with a median correlation coefficient of 0.97 and 0.95 for ECFP4 and MACCS, respectively. SHAP mean values were overall more strongly correlated for ECFP4- than MACCS-based models.

Taken together, the results in Figures 6 and 7 revealed that SHAP values of features originating from models built using ML algorithms were highly correlated, showing that the different methods prioritized similar chemical patterns for predictions that were consistently detected in the basis of SHAP values.

Visualization of SHAP Values. To rationalize model predictions, features with highest SHAP values for individual predictions were extracted, first for the simplistic MACCS fingerprint that encodes the presence or absence of predefined chemical patterns. Figure 8 shows MACCS feature weights for the correct prediction of three compounds using SVM, RF, and DNN models. The first compound (Figure 8a) was an antiapoptotic Bcl-2 inhibitor and the second (Figure 8b) a melanin-concentrating hormone receptor 1 antagonist. The third compound (Figure 8c) was a factor X inhibitor. For each ML model and test compound, SHAP values for MACCS features are reported in a separate graph. Positive and negative feature contributions are identified using sequential red and blue arrows, respectively. The length of each arrow is proportional to the SHAP value for a given feature, and the MACCS keys corresponding to the top-ranked variables with

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

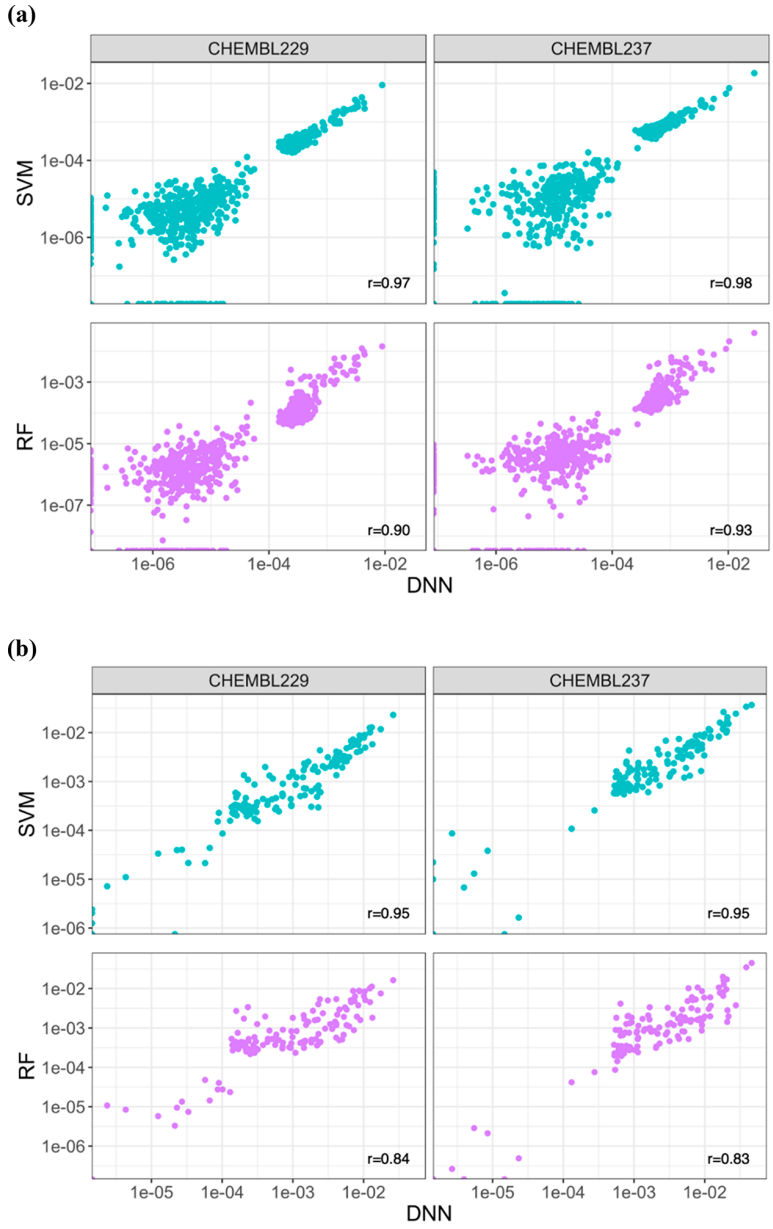


Figure 6. Comparison of SHAP values. Mean absolute SHAP values for features originating from different ML models are compared in a pairwise manner. Each data point represents a pairwise value comparison for a given feature. Different ML models were generated on the basis of (a) ECFP4 and (b) MACCS.

(largest absolute SHAP value) are given. The expected value is obtained as the average model output over training set instances and corresponds to the predicted probability of a test compound with unknown feature values. It is also referred to as the base probability. SHAP values quantify the influence of a given feature on a prediction and modify the expected value. When SHAP values are added to this base value, the output probability of the original ML model is obtained (shown in bold).

The three compounds in Figure 8 were correctly classified as actives by the three ML algorithms. Moreover, different methods shared most top-ranked features, indicating that similar chemical patterns determined the prediction of a given compound. However, the absolute importance values differed between ML methods and other features with smaller SHAP values also contributed to the predictions, resulting in different final output probabilities. DNN models produced the highest output probabilities of activity for these test compounds, whereas RF models gave the smallest ones.

Figure 9 shows feature weights for three other exemplary compounds, which were represented by ECFP4, including a κ opioid receptor (Figure 9a), melanocortin receptor 1B (Figure 9b), and orexin receptor 2 (Figure 9c) ligands. ML models correctly predicted these active compounds and SHAP values were calculated to examine the prioritized features for the predictions. In this case, positive and negative feature contributions were displayed using the ECFP4 feature index (obtained after fingerprint folding). Again, most highly weighted features were common to SVM, RF, and DNN models. RF gave the overall lowest output probability, due to

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

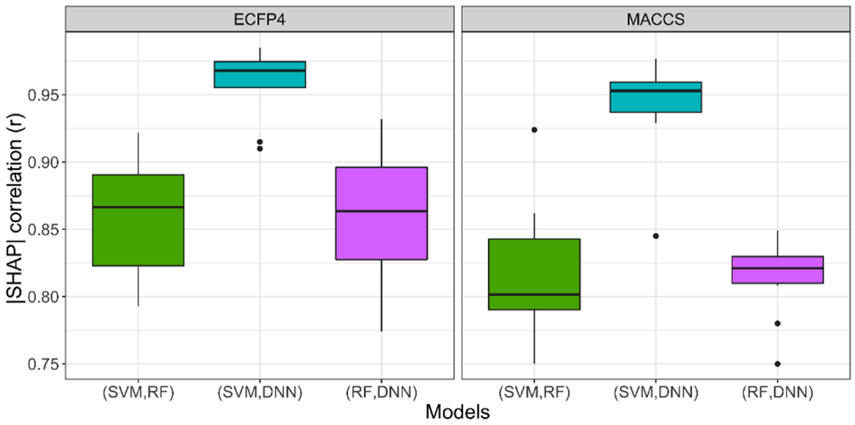


Figure 7. SHAP value correlation between ML models. For each activity class, the correlation between mean absolute SHAP values was calculated for different ML models. Boxplots report the pairwise correlation of SVM vs RF (green), SVM vs DNN (blue), and RF vs DNN (purple) for two molecular representations (ECFP4, left; MACCS, right).

smaller individual feature weights. The corresponding substructures of top-ranked important features shared by the three algorithms were mapped onto the compounds. In Figure 9a, feature with index #566 is highlighted, which was relevant for the three models applying a SHAP threshold value of ~0.07. Feature #566 was ranked top-1 (SVM), -2 (RF), and -2 (DNN). On the other hand, in Figure 9b, the highlighted feature #637 was ranked top-5 (SVM), -1 (RF), and -6 (DNN). However, in the latter case, features ranked higher than #637 (SMILES, [CH2]CNC(C)=O) represented substructures of #637 (#1010, CC; #29, [CH2]NC; #118, [CH2]NC(C)=O; #236, CC([NH])=O; #960, [CH2]C-[NH]) and were thus correlated. Finally, in Figure 9c, highlighted features include #843 (ranked top-1 (SVM), -1 (RF), and -2 (DNN)) and #268 (ranked top-2 (SVM), -3 (RF), and -1 (DNN)). The SHAP values representations in Figures 8 and 9 provide global explanations for a given prediction and enable comparison of feature importance across different models and methods.

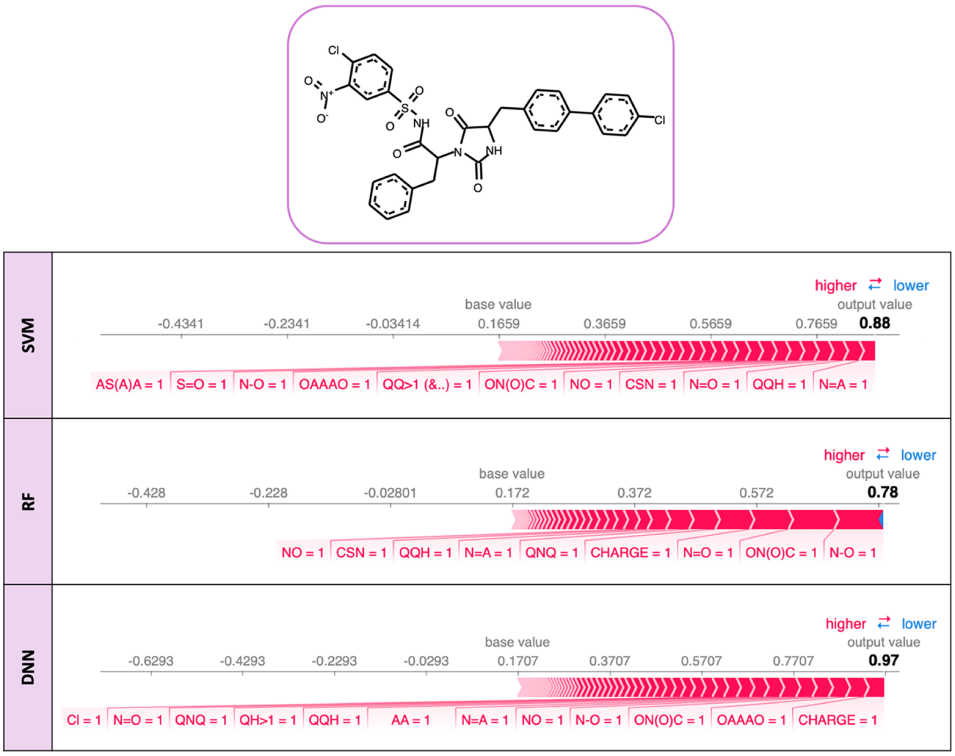
Feature Mapping onto Compounds. The use of the ECFP4 fingerprint made it possible to map highly weighted topological features onto molecular graphs and analyze resulting substructures. Although ECFP4 folding might lead to feature "collisions" (i.e., different atom environments might be encoded by the same element), such collisions were only very rarely observed for individual compounds because of their generally low number of hash values compared to the size of the folded fingerprint. In global model interpretation, a unique weight is obtained for each feature. SHAP values explain individual predictions, and for a given compound, correspondence between a given feature and substructure is generally unequivocal. Furthermore, different mapped features might contribute to the formation of coherent, overlapping, or distinct substructures. Figure 10 provides an example for the rationalization of a prediction on the basis of SHAP values. Figure 10a depicts the mapping of the most relevant features onto a compound active against the κ opioid receptor, and Figure 10b gives an overview of the positive and negative feature contributions. All three ML models correctly predicted this test compound, and the substructures resulting from mapping of features that determined these predictions were explored. For feature mapping, a threshold should be defined that can be based on the absolute SHAP value, the signed value (accounting for positive or negative contributions), or the number of top-ranked features. Therefore, depending on the application, different types of threshold values can be used. In this case, the threshold was iteratively varied, and results for different SHAP threshold values are shown in the figure.

In Figure 10a, the top-1 and -2 ranked features from SVM, RF, and DNN models are highlighted. For the three models, mapping of important features lineated the same or similar substructures. Figure 10b provides a complementary view of cumulative positive or negative feature contributions. In this case, RF and DNN models predicted a lower probability of activity (p of ~0.60) than the SVM model (p = 0.97), which largely resulted from negative feature contributions, especially for DNN, which were absent in the SVM model. SHAP results suggest that RF and DNN models made use of the absence of some features to discriminate between active and inactive training compounds. However, such prioritization had a negative impact on the model output for this exemplary active test compound, leading to a lower output probability. Accordingly, a noninformative bias in the training set was likely exploited by these two ML models. For example, both models penalized the absence of feature #12 (SMARTS pattern: [#6D4v4+0H0R], SMILES: C), which was present in 91% of the positive and only in the 8% of the negative training compounds. The representation also shows that the majority of features with positive contributions to the prediction of activity were conserved.

Comparison of Structural Analogs. Analog series provide interesting test cases for local model diagnostics. In most cases, analogs from the same series are predicted to be active because of their high structural similarity. However, there can be exceptions where small structural differences between compounds abruptly change the predicted probability of activity. Such incorrect predictions are of particular interest to better understand intrinsic limitation of activity predictions, provided the underlying models can be interpreted. Figure 11 presents the SHAP analysis of SVM predictions for two histamine H3 receptor antagonists with comparable potency (having pK_i values of 6.2 and 6.3, respectively). One was predicted correctly, the other incorrectly. Figure 11a shows the ECFP4 features with the highest positive and negative contributions on predicted activity. The first analog was accurately predicted (p = 0.98), but the second was not (p =

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

(a)



(b)

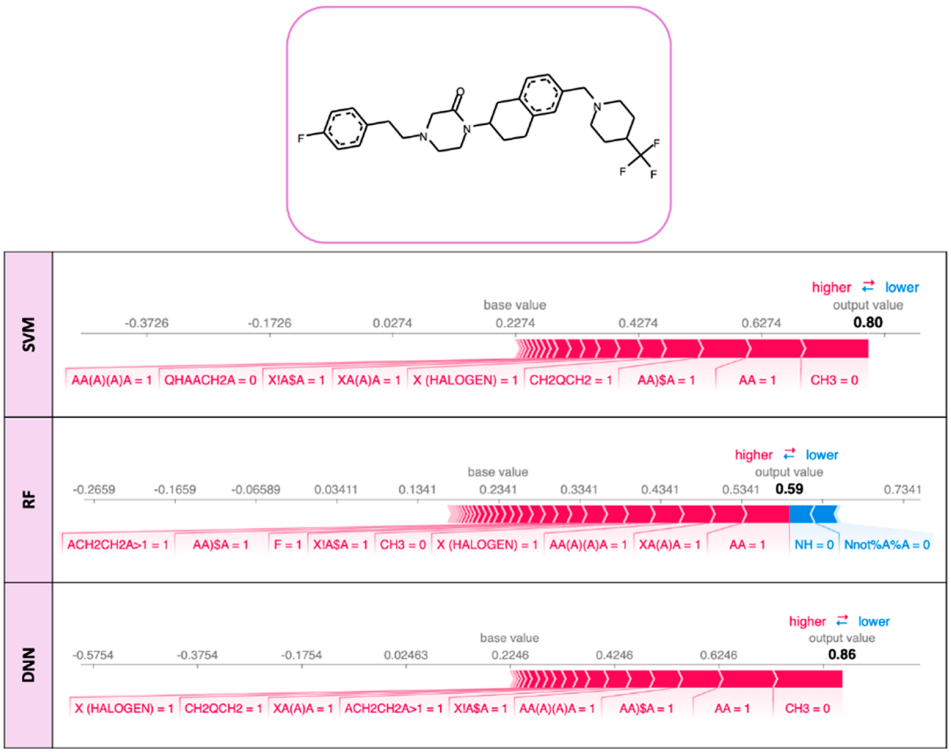


Figure 8. continued

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

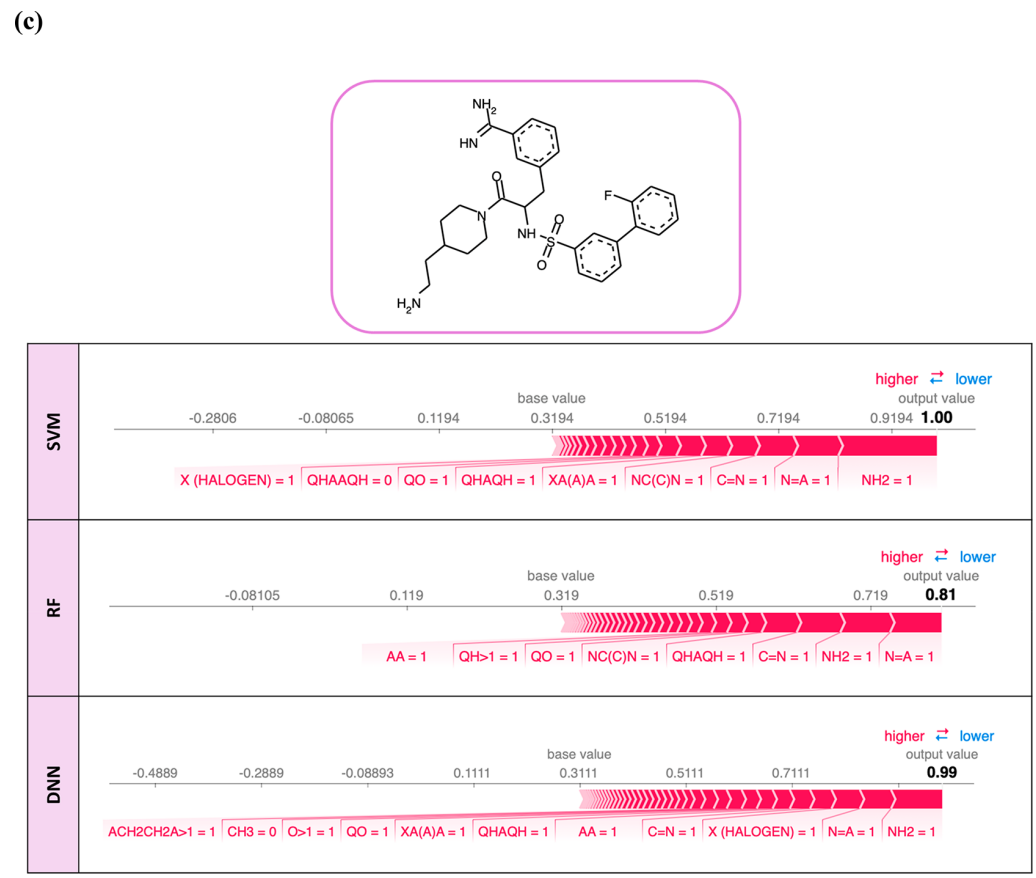


Figure 8. SHAP values for MACCS keys. Shown are two exemplary test compounds that were represented using MACCS keys and correctly predicted by SVM, RF, and DNN models including a (a) Bcl-2 inhibitor, (b) melanin-concentrating hormone receptor 1 antagonist, and (c) factor X inhibitor. SHAP positive (red) and negative (blue) feature weights are given for the three models. The expected base and output value (bold) is also shown. The following symbols are used: A, any element symbol; Q, heteroatom; X, other than H, C, N, O, Si, P, S, F Cl, Br, I; $, ring bond; !, aliphatic bond; %, aromatic bond.

0.19). The substructure formed by features with the highest positive contribution was shared by both compounds but obtained a larger SHAP value for the first analog. Moreover, the correctly predicted compound did not yield any feature with a negative contribution. By contrast, for the incorrectly predicted analog, features making a large negative contribution were identified. Consequently, the substructure formed by features with largest negative contribution according to the SVM model was only present in the incorrectly predicted analog. Figure 11b reports the base and SHAP values for the two compound predictions. Even though most of the variables with positive contributions were shared by both compounds, the second analog exhibited a number of features that negatively impacted the prediction. Thus, SHAP analysis uncovered a model error and made it possible to rationalize why these two analogs produced different model outputs. On the basis of such insights, it can be attempted to further optimize SVM models for individual predictions.

Global Model Diagnostics. SHAP analysis can conveniently be used as a global model diagnostic by comparing decisions of different ML models on the same compound data set, which aids in model selection. Moreover, consensus features can be identified across methodologically distinct models that can be selected for practical applications. Figure 12 presents an example of SHAP-based model comparison and selection. Figure 12a shows a score plot of predicted probabilities of activity for compounds using DNN and SVM models. Red dots in the upper-right panel represent active compounds that are correctly predicted by both methods, and blue dots in the bottom-left panel are inactive compounds correctly detected by SVM and DNN. The compounds falling into other regions of the plot have been incorrectly predicted by only one of the methods. An exemplary active compound that is correctly predicted by DNN but not by SVM is indicated. In Figure 12b, the SHAP contribution plots are shown for this compound and the SVM and DNN models. It is evident that many features were equally weighted using SHAP for predictions with both models. However, SVM was found to assign negative contributions to a number of atom environments that were not considered by DNN. To further reduce the black-box character of these model predictions, highly weighted features were mapped onto this compound, as depicted in Figure 12c. The SHAP threshold was adjusted such that top-1 as well as -3 ranked features with positive contributions were obtained from both SVM and DNN models. For SVM, the top-ranked features with negative contributions were also selected. Such features were absent in the DNN model, as discussed above. Figure 12c shows that features important for the prediction of activity mapped to the same region in the molecule. However, SVM also negatively weighted similar parts of the compound formed by overlapping atom environments, thus reducing the output probability.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

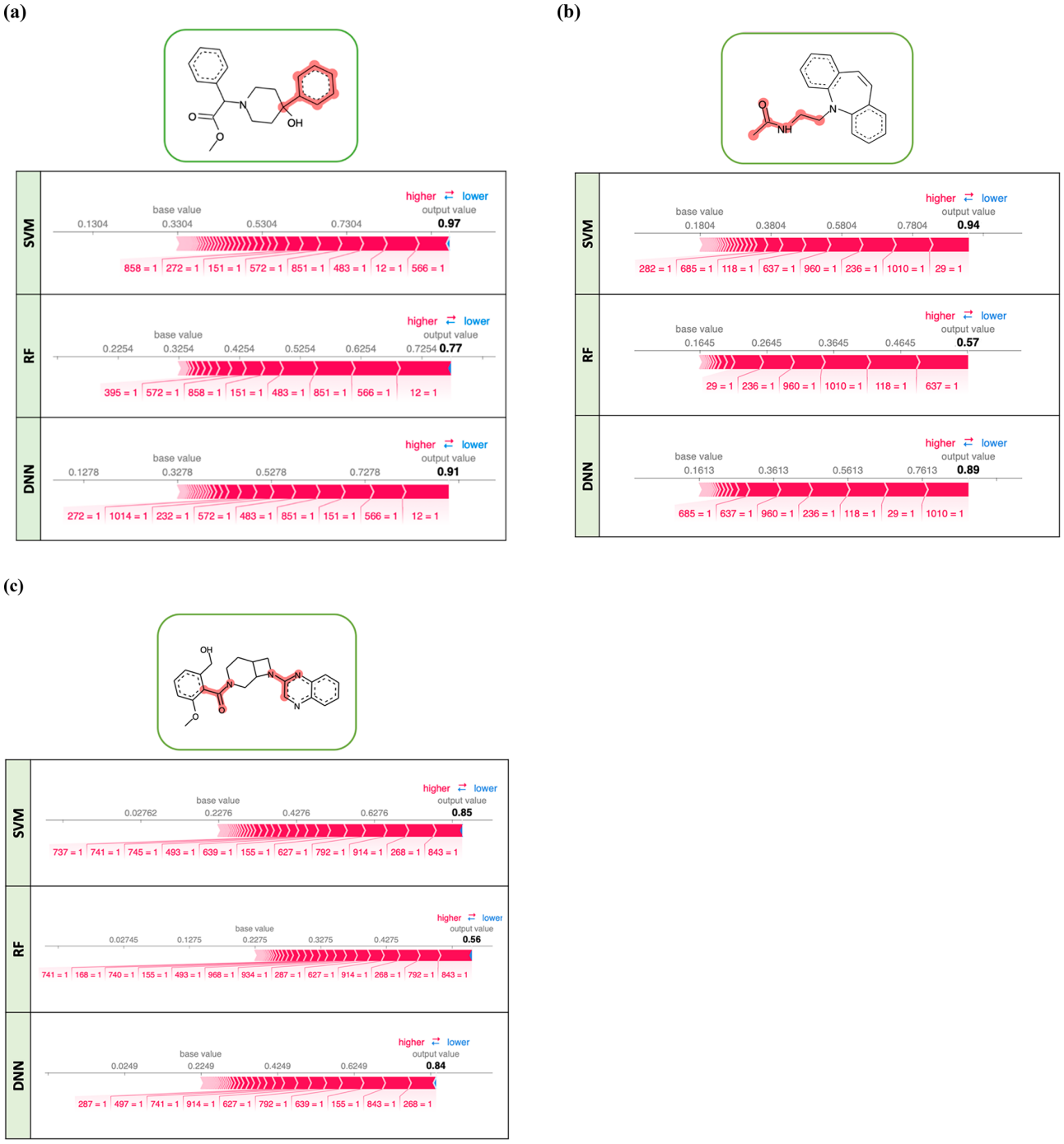


Figure 9. SHAP values for ECFP4 features. Shown are two exemplary test compounds that were represented using ECFP4 and correctly predicted by SVM, RF, and DNN models including a ligand of the (a) κ opioid receptor, (b) melanocortin receptor 1B, and (c) orexin receptor 2. The representation is according to Figure 8. In addition, top-ranked features are highlighted in compound structures.

Thus, in this case, the model diagnostic detected SVM-dependent inconsistencies in feature prioritization, which were absent in the DNN model. On the basis if these observations, the DNN model would be prioritized.

CONCLUSIONS

In this work, the SHAP method has been introduced for the interpretation of compound activity predictions using ML models, regardless of their complexity. As an ML model diagnostic, SHAP is generally applicable to ML models including ensemble and DL models, which makes it possible to shed light on their black-box nature. SHAP values quantify feature importance for ML in a consistent manner. Furthermore, the SHAP analysis scheme introduced herein provides visual access to feature importance and enables structural interpretation of ML predictions including DNNs. By application of the SHAP methodology, variables with increasing influence on predictions can be explored and detect

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

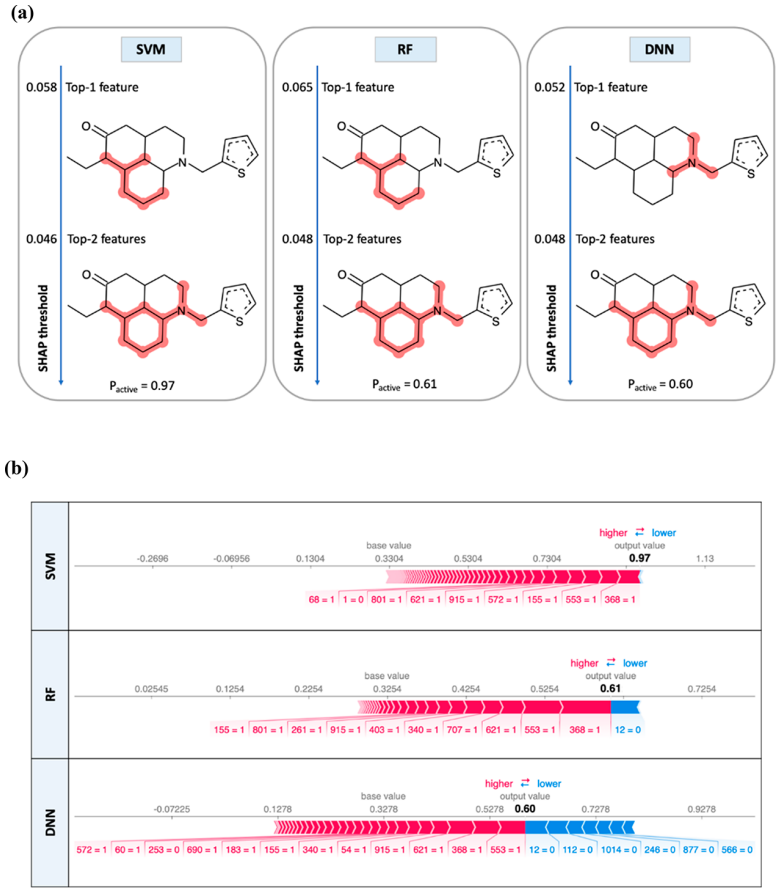


Figure 10. SHAP visualization for ECFP4. SHAP results are shown for an exemplary κ opioid receptor antagonist. In (a), the probability of activity predicted by SVM (left), RF (center), and DNN (right) is reported at the bottom of the boxes and the most important features for determining these predictions (top-1 and top-2) according to SHAP analysis are mapped onto the compound and highlighted. For top-ranked features, the corresponding SHAP values are reported. In (b), positive (red) and negative (blue) feature contributions are shown for SVM (top), RF (middle), and DNN (bottom). The output value (bold) corresponds to the output probability of each ML model.

potential sources of bias of predictions or confirm their consistency and further validate a model. It is important to consider the applicability domain of explanatory methods because interpretations will be strongly influenced by training data and conditions. Here, it is important to note that the SHAP methodology is applicable to essentially all ML approaches including regression techniques. For ML methods and especially in the context of DL, SHAP offers novel opportunities for the rationalization of predictive models and for reducing or eliminating their black-box character. In future work, SHAP analysis might be further extended to better understand multitask learning for compound activity prediction.

EXPERIMENTAL SECTION

1. Compound Data Sets. Machine learning inevitably depends on compounds from the literature and their reported activity data. ChEMBL is the primary repository for active compounds from the medicinal chemistry literature.[44] From ChEMBL version 24, 10 activity classes were selected for ML.

For each selected compound, literature reference(s) and the presence of direct interactions (i.e., assay relationship type "D") with a human single-protein target at the highest confidence level (i.e., assay confidence score 9) were required. As potency measurements, explicitly specified (assay-independent) equilibrium constants (K_i values) were required. Activity measurements provided in ChEMBL were taken from original publications. When multiple K_i values were available for a compound and fell within the same order of magnitude, the mean value was determined. If differences between measurements exceeded 1 order of magnitude, the compound was discarded. Only compounds with (mean) pK_i of at least 5 were ultimately selected to exclude borderline active compounds from further consideration. Furthermore, compounds with potentially inconsistent activity records including comments such as "inactive", "inconclusive", or "not active" were discarded. Taken together, these criteria exclusively select compounds with highest ChEMBL confidence scores and highest activity data confidence.[45] In addition, all compounds meeting high-confidence selection criteria were screened for pan-assay interference compounds (PAINS)[46] using substructure libraries from public filters[44,47,48] and compounds with PAINS alerts were discarded (less than 1%).

Selected data sets were required to contain at least 200 compounds belonging to at least 50 different analog series computationally determined[49] on the basis of matched molecular pair (MMP) relationships.[50] Selection of activity classes consisting of large numbers of analog series ensured the presence of defined subsets of structurally analogous compounds that were distinct from others. Activity classes of sufficient size and intraclass structural diversity were essential for meaningful ML-based activity modeling. Since this study aimed to detect chemical features determining activity predictions, confirmed activity of compounds against a given target based on high-confidence activity data was another key criterion for an activity class. Table 1 specifies selected classes, which consisted of 243−955 compounds and 57−216 analog series, respectively. To prevent

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

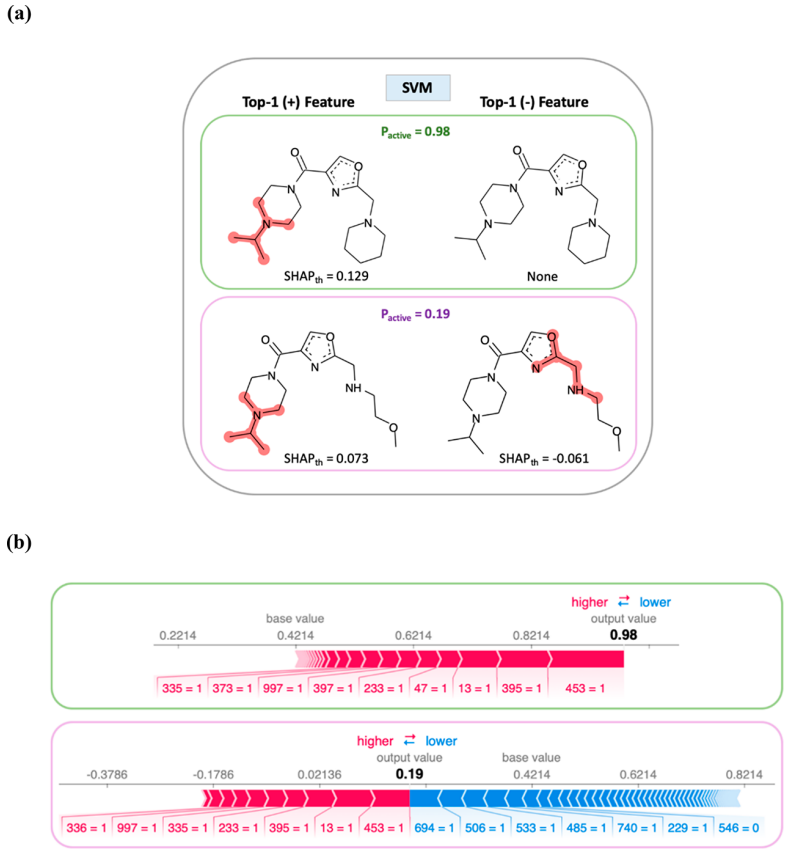


Figure 11. Rationalizing SVM predictions for two analogs. (a) Two analogs are shown (with ECFP4 Tanimoto similarity of 0.6), and features with the largest positive and negative contributions to SVM predictions are highlighted. $SHAP_{th}$ indicates the SHAP threshold value for the top-1 ranked feature (such that only this feature is obtained). The analogs have different predicted probabilities of activity (P_{active}). (b) For the analogs, features with positive (red) and negative (blue) SHAP values are visualized.

potential structural bias of predictions,[51,52] analogs from different series were selected as positive (active) training and test instances. Training sets contained 70% of the analog series per activity class and corresponding test sets 30% of the series. On average, training and test sets included 366 (157 to 683) and 163 (70−278) active compounds, respectively. As negative (inactive) training and test instances, compounds were randomly selected from ZINC,[48] i.e., consistently 1000 compounds per training and test set.

2. Molecular Representations. Extended connectivity fingerprint with bond diameter 4 (ECFP4)[53] is a topology descriptor encoding layered atom environments as numeric identifiers using a hashing function. SMARTS patterns corresponding to each atom environment (codified by a hash value) were stored. Therefore, ECFP4 features can be mapped back onto the compounds. This feature set fingerprint is variable in size, but a constant-length 1024-bit representation was obtained through modulo mapping. In addition, MACCS structural keys[54] were used in a binary fingerprint format encoding the presence (bit set on) or absence (off) of 166 predefined structural patterns or fragments. The OEChem toolkit[55] and in-house Python scripts were used for fingerprint calculations.

3. Machine Learning Models. *3.1. Support Vector Machine.* The SVM classifier finds a hyperplane in a multidimensional space that maximizes the distance between the support vectors of each class, known as *margin*.[17] The support vectors are the training instances of one class that are closest to the other class. SVM enables nonlinear modeling through the application of the *kernel trick*,[56] i.e., the use of kernel functions to map training compounds into a higher-dimensional feature space representation in which the classes might be linearly separable. For compound classification, the nonlinear Tanimoto kernel[54] is one of the best performing kernel functions.[57,58] The SVM implementation of scikit-learn[56] with customized Tanimoto kernel was used for all calculations.

3.2. Random Forest. RF is an ensemble of decision trees (DTs) that aims at reducing the variance of individual trees.[16] RF is based on bootstrap aggregating according to which training DTs with distinct compound subsets are generated. In addition, a random subset of features is used to minimize correlations between DTs. The final RF prediction results from a consensus across the DT population. RF calculations were carried out with scikit-learn.[59]

3.3. Feedforward Deep Neural Networks. A DNN is a series of functional transformations (neurons) that learn how to modify input values to obtain a desired output.[60] Accordingly, DNNs have an input layer, multiple hidden layers, and an output layer. First, a neuron's input values (x_1, ..., x_D) are linearly combined considering a set of weights ($\mathbf{w}$) and biases (b). Then, a differentiable nonlinear activation function (h) is applied to obtain the neuron's output (y_j) according to eq 4:[61]

$$y_j = h\left(\sum_{i=1}^{D} \omega_{ji}^{(n)} x_i + b_j^{(n)}\right) \quad (4)$$

where n indicates the layer number. Training aims at determining the weights and biases that minimize the cost function (e.g., cross-entropy).[21] Gradient descent is applied to update weights by considering small steps (defined by the learning rate) in the direction of the negative gradient and can be efficiently calculated using backpropagation.[60] DNNs were generated using TensorFlow[61] and Keras.[62]

3.4. Hyperparameter Optimization. Model hyperparameters were optimized through internal 2-fold cross-validation and grid search.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

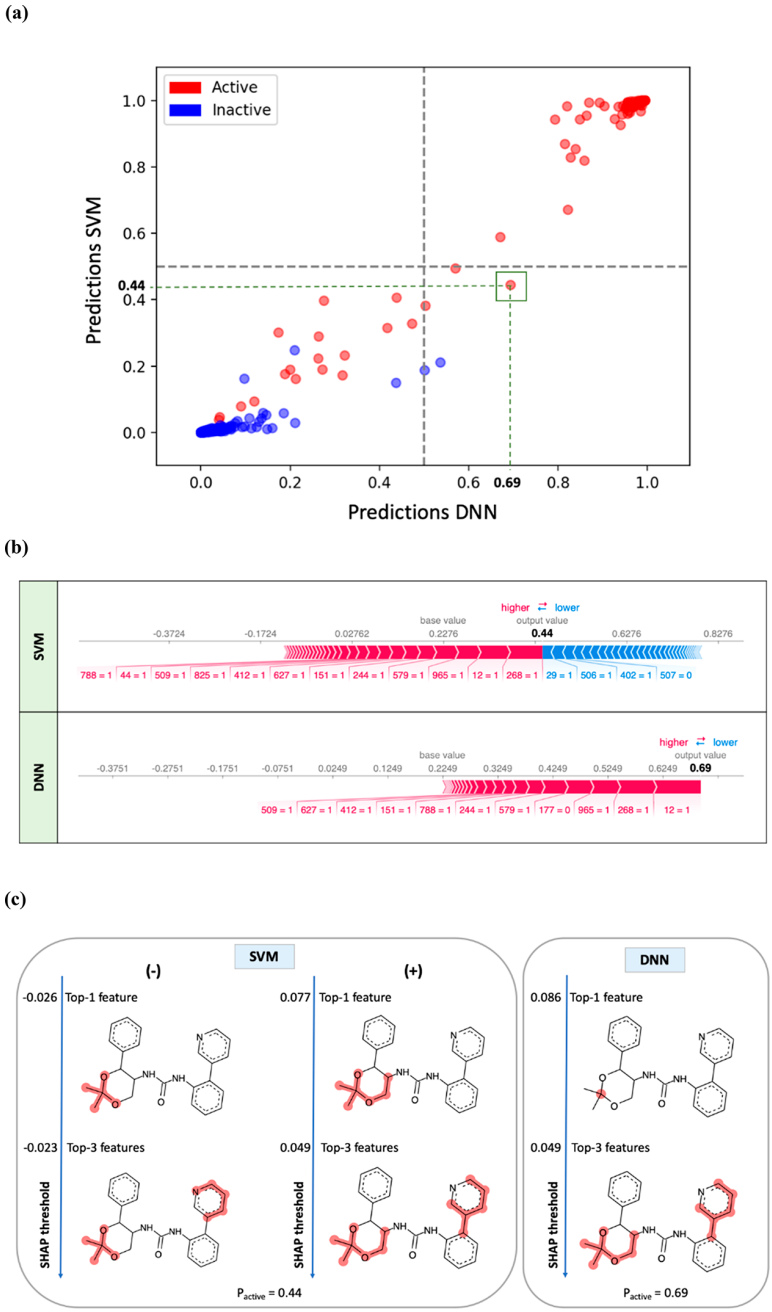


Figure 12. Interpretation of DNN and SVM predictions. (a) Score plots show the output probabilities of activity against orexin receptor 2 for DNN and SVM models. A green square marks an exemplary compound that is incorrectly classified by SVM (p = 0.44) but correctly predicted by DNN (p = 0.69). (b) Plots for SVM and DNN report SHAP feature values that modify the base value (0.22), with a positive (red) or negative (blue) sign, to yield the final output probability (bold). For the DNN model, features with negative contributions to the output probability were absent. (c) ECFP4 features with largest positive and negative SHAP values are shown for the SVM model (i.e., the top-1 feature and the top-3 ranked features). For the DNN model, only the features with positive SHAP values are available.

The same randomized data splits were considered for training (80%) and internal validation (20%) for different ML methods.[63] Best hyperparameters were selected according to area under the ROC curve (AUC) optimization (average across folds).

For SVM, the regularization term C was optimized with candidate values of 0.01, 0.1, 1, and 10. In addition, SVM models were built with and without class weights.[58] The use of class weights consists in penalizing errors on the minority class more than errors on the majority class.

For RF models, the number of trees was consistently set to 500 and three numerical hyperparameters were optimized including the minimum number of samples required to split a leaf node (1, 5, 10) or an internal node (2, 8, 16) and the maximum number of features considered when searching for the best split (i.e., square root, $\log_2$). Furthermore, models were built with and without class weights.

Different network architectures were tested for DNN models, with the following number of neurons in hidden layers: [100,500], [200,100], [2000,1000], [200,100,100], and [2000,1000,100]. The activation function was Rectified Linear Unit (ReLU) except at the

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

output layer, where a sigmoid function was applied. In addition, three initial learning rates (0.1, 0.01, 0.001) were tested and values were reduced when reaching a loss plateau. L2 regularization and drop-out (25% or 50%) were applied to all hidden layers. Three batch sizes (64, 128, 256) were tested, Adam was used as the optimization function, and the number of epochs was set to 50 and 200 during internal and external validation, respectively.

3.5. Performance Measures. Predictive performance on test sets was evaluated using three metrics: AUC, balanced accuracy (BA),[64] and Matthew's correlation coefficient (MCC).[65] BA and MCC are defined by eqs 5 and 6, respectively.

$$\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR}) \tag{5}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{6}$$

To statistically compare MCC values before and after feature elimination, nonparametric Wilcoxon tests[66] were carried out.

4. Feature Contributions. Feature contributions were assessed following the SHAP approach detailed in the Results sections. The feature contributions represented by Shapley values are meant to satisfy three axioms including *local accuracy*, *consistency*, and *nonexistence* (or null effect).[67,68]

5. Data Availability. Compound activity classes used here are made available in an open access deposition on the ZENODO platform.[69]

AUTHOR INFORMATION

Corresponding Author

*Phone: +49-228-7369-100. Fax: +49-228-7369-101. E-mail: bajorath@bit.uni-bonn.de.

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.-P.) from the European Union's Horizon 2020 Research and Innovation program under the Marie Skłodowska-Curie Grant Agreement 676434, "Big Data in Chemistry" ("BIG-CHEM", http://bigchem.eu). The article reflects only the authors' view, and neither the European Commission nor the Research Executive Agency (REA) is responsible for any use that may be made of the information it contains. The authors thank OpenEye Scientific Software, Inc., for providing a free academic license for the OpenEye toolkit, and Scott Lundberg for the SHAP library. The authors also thank Dagmar Stumpfe and Martin Vogt for help with compound data analysis.

ABBREVIATIONS USED

AUC, area under the ROC curve; BA, balanced accuracy; DL, deep learning; DNN, deep neural network; DT, decision tree; ECFP, extended connectivity fingerprint; LIME, local interpretable model-agnostic explanations; MCC, Matthew's correlation coefficient; ML, machine learning; PAINS, pan-assay interference compounds; (Q)SAR, (quantitative) structure−activity relationship; RF, random forest; SHAP, Shapley additive explanations; SVM, support vector machine

REFERENCES

(1) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Cheminformatics: Quo Vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413−1437.

(2) Rodríguez-Pérez, R.; Miyao, T.; Jasial, S.; Vogt, M.; Bajorath, J. Prediction of Compound Profiling Matrices Using Machine Learning. *ACS Omega* **2018**, *3*, 4713−4723.

(3) Lavecchia, A. Machine-Learning Approaches in Drug Discovery: Methods and Applications. *Drug Discovery Today* **2015**, *20*, 318−331.

(4) Lo, Y.; Rensi, S. E.; Torng, W.; Altman, R. B. Machine Learning in Chemoinformatics and Drug Discovery. *Drug Discovery Today* **2018**, *23*, 1538−1546.

(5) Cherkasov, A.; Muratov, E.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. QSAR Modeling: Where Have You Been? Where Are You Going to? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(6) Zakharov, A. V.; Peach, M. L.; Sitzmann, M.; Nicklaus, M. C. QSAR Modeling of Imbalanced High-Throughput Screening Data in PubChem. *J. Chem. Inf. Model.* **2014**, *54*, 705−712.

(7) Lewis, R. A. A General Method for Exploiting QSAR Models in Lead Optimization. *J. Med. Chem.* **2005**, *48*, 1638−1648.

(8) Bajorath, J. Integration of Virtual and High-throughput Screening. *Nat. Rev. Drug Discovery* **2002**, *1*, 882−894.

(9) Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved Naïve Bayesian Modeling of Numerical Data for Absorption, Distribution, Metabolism and Excretion (ADME) Property Prediction. *J. Chem. Inf. Model.* **2006**, *46*, 1945−1956.

(10) Guha, R. On the Interpretation and Interpretability of Quantitative Structure-Activity Relationship Models. *J. Comput.-Aided Mol. Des.* **2008**, *22*, 857−871.

(11) Doweyko, A. M. QSAR: Dead or Alive? *J. Comput.-Aided Mol. Des.* **2008**, *22*, 81−89.

(12) Sieg, J.; Flachsenberg, F.; Rarey, M. In the Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 947−961.

(13) Polishchuk, P. Interpretation of Quantitative Structure-Activity Relationship Models: Past, Present, and Future. *J. Chem. Inf. Model.* **2017**, *57*, 2618−2639.

(14) Hansen, K.; Baehrens, D.; Schroeter, T.; Rupp, M.; Müller, K.-R. Visual Interpretation of Kernel-Based Prediction Models. *Mol. Inf.* **2011**, *30*, 817−826.

(15) Balfer, J.; Bajorath, J. Introduction of a Methodology for Visualization and Graphical Interpretation of Bayesian Classification Models. *J. Chem. Inf. Model.* **2014**, *54*, 2451−2468.

(16) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.

(17) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.

(18) Balfer, J.; Bajorath, J. Visualization and Interpretation of Support Vector Machine Activity Predictions. *J. Chem. Inf. Model.* **2015**, *55*, 1136−1147.

(19) Hu, Y.; Bajorath, J. Learning from 'Big Data': Compounds and Targets. *Drug Discovery Today* **2014**, *19*, 357−360.

(20) Papadatos, G.; Gaulton, A.; Hersey, A.; Overington, J. P. Activity, Assay and Target Data Curation and Quality in the ChEMBL database. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 885−896.

(21) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015.

(22) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *25*, 1097−1105.

(23) Hinton, G. E.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. Deep Neural Networks for Acoustig Modeling in Speech Recognition. *IEEE Signal Process. Mag.* **2012**, *29*, 82−97.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

(24) Baskin, I.; Winkler, D.; Tetko, I. V. A Renaissance of Neural Networks in Drug Discovery. *Expert Opin. Drug Discovery* **2016**, *11*, 785−795.
(25) Ekins, S. The Next Era: Deep Learning in Pharmaceutical Research. *Pharm. Res.* **2016**, *33*, 2594−2603.
(26) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.
(27) 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. Cheminf.* **2017**, *9*, No. e45.
(28) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.; Hochreiter, S. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441−5451.
(29) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.
(30) Xu, Y.; Ma, J.; Liaw, A.; Sheridan, R. P.; Svetnik, V. Demystifying Multitask Deep Neural Networks for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model.* **2017**, *57*, 2490−2504.
(31) Rodríguez-Pérez, R.; Bajorath, J. Prediction of Compound Profiling Matrices, Part II: Relative Performance of Multitask Deep Learning and Random Forest Classification on the Basis of Varying Amounts of Training Data. *ACS Omega* **2018**, *3*, 12033−12040.
(32) Lundberg, S. M.; Nair, B.; Vavilala, M. S.; Horibe, M.; Eisses, M. J.; Adams, T.; Liston, D. E.; Low, D. K.; Newman, S.; Kim, J.; Lee, S. Explainable Machine-learning Predictions for the Prevention of Hypoxaemia During Surgery. *Nat. Biomed. Eng.* **2018**, *2*, 749−760.
(33) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Support Vector Machine Classification and Regression Prioritize Different Structural Features for Binary Compound Activity and Potency Value Prediction. *ACS Omega* **2017**, *2*, 6371−6379.
(34) Hastie, T.; Tibshirani, R.; Friedman, J. H. *The Elements of Statistical Learning*; Springer: Berlin, 2009.
(35) Iooss, B.; Saltelli, A. Introduction to Sensitivity Analysis. In *Handbook of Uncertainty Quantification*; Ghanem, R., Higdon, D., Owhadi, H., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp 1−20.
(36) So, S. S.; Richards, W. G. Application of Neural Networks: Quantitative Structure-Activity Relationships of the Derivatives of 2,4-Diamino-5-(Substituted-Benzyl)Pyrimidines as DHFR Inhibitors. *J. Med. Chem.* **1992**, *35*, 3201−3207.
(37) Baskin, I. I.; Ait, A. O.; Halberstam, N. M.; Palyulin, V. A.; Zefirov, N. S. An Approach to the Interpretation of Backpropagation Neural Network Models in QSAR Studies. *SAR QSAR Environ. Res.* **2002**, *13*, 35−41.
(38) Marcou, G.; Horvath, D.; Solov'ev, V.; Arrault, A.; Vayer, P.; Varnek, A. Interpretability of SAR/QSAR Models of Any Complexity by Atomic Contributions. *Mol. Inf.* **2012**, *31*, 639−642.
(39) Fujita, T.; Winkler, D. A. Understanding the Roles of the "Two QSARs". *J. Chem. Inf. Model.* **2016**, *56*, 269−274.
(40) Johansson, U.; Sönströd, C.; Norinder, U.; Boström, H. Trade-Off between Accuracy and Interpretability for Predictive in Silico Modeling. *Future Med. Chem.* **2011**, *3*, 647−663.
(41) Lundberg, S. M.; Lee, S. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; NIPS, 2017.
(42) Ribeiro, M. T.; Singh, S.; Guestrin, C. "Why Should I Trust You?"Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2016**, 1135−1144.
(43) Shapley, L. S. A Value for N-Person Games. In *Contributions to the Theory of Games*; Kuhn, H. W., Tucker, A. W., Eds.; Annals of Mathematical Studies; Princeton University Press, 1953; pp 307−317.
(44) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.
(45) Hu, Y.; Bajorath, J. Influence of Search Parameters and Criteria on Compound Selection, Promiscuity, and Pan Assay Interference Characteristics. *J. Chem. Inf. Model.* **2014**, *54*, 3056−3066.
(46) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53*, 2719−2740.
(47) RDKit: Cheminformatics and Machine Learning Software. 2013. http://www.rdkit.org (accessed June 3, 2019).
(48) Sterling, T.; Irwin, J. J. ZINC 15–Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.
(49) Stumpfe, D.; Dimova, D.; Bajorath, J. Computational Method for the Systematic Identification of Analog Series and Key Compounds Representing Series and Their Biological Activity Profiles. *J. Med. Chem.* **2016**, *59*, 7667−7676.
(50) Kenny, P. W.; Sadowski, J. Structure Modification in Chemical Databases. In *Cheminformatics in Drug Discovery*; Oprea, T. I., Ed.; Wiley-VCH: Weinheim, Germany, 2004; pp 271−285.
(51) Scior, T.; Bender, A.; Tresadern, G.; Medina-Franco, J. L.; Martínez-Mayorga, K.; Langer, T.; Cuanalo-Contreras, K.; Agrafiotis, D. K. Recognizing Pitfalls in Virtual Screening: A Critical Review. *J. Chem. Inf. Model.* **2012**, *52*, 867−881.
(52) Rodríguez-Pérez, R.; Bajorath, J. Multitask Machine Learning for Classifying Highly and Weakly Potent Kinase Inhibitors. *ACS Omega* **2019**, *4*, 4367−4375.
(53) Rogers, D.; Hahn, M. Extended Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(54) *MACCS Structural Keys*; Accelrys: San Diego, CA, 2011.
(55) *OEChem TK*, version 2.0.0; OpenEye Scientific Software: Santa Fe, NM, 2015.
(56) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the 5th Annual Workshop on Computational Learning Theory; Pittsburgh, Pennsylvania, 1992*; ACM: New York, 1992; pp 144−152.
(57) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Netw* **2005**, *18*, 1093−1110.
(58) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.
(59) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(60) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.
(61) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M., Yu, Y.; Zheng, X. TensorFlow: A System for Large-scale Machine Learning. Presented at the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, 2016.
(62) Chollet, F. Keras, version 2.1.3, 2015. https://github.com/keras-team/keras (accessed January 17, 2018).
(63) Baumann, D.; Baumann, K. Reliable Estimation of Prediction Errors for QSAR Models under Model Uncertainty Using Double Cross-Validation. *J. Cheminf.* **2014**, *6*, No. e47.
(64) Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; Buhmann, J. M. The Balanced Accuracy and Its Posterior Distribution. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)* **2010**, 3121−3124.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

(65) Matthews, B. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta, Protein Struct.* **1975**, *405*, 442−451.
(66) Conover, W. J. On Methods of Handling Ties in the Wilcoxon Signed-Rank Test. *J. Am. Stat. Assoc.* **1973**, *68*, 985−988.
(67) Osborne, M. J.; Rubinstein, A. *A Course in Game Theory*; The MIT Press: Cambridge, MA, 1994.
(68) Young, H. P. Monotonic Solutions of Cooperative Games. *Int. J. Game Theory.* **1985**, *14*, 65−72.
(69) Rodríguez-Pérez, R.; Bajorath, J. Compound Activity Classes from ChEMBL for Machine Learning Analysis. https://zenodo.org/record/3362353#.XUrdh5MzafU.

DOI: 10.1021/acs.jmedchem.9b01101
J. Med. Chem. XXXX, XXX, XXX−XXX

Summary

In this chapter, a conceptually new interpretation method to understand activity predictions from any ML model was presented. Shapley additive explanations (SHAP) represents an extension of local interpretable model-agnostic methods where feature weights are approximated as Shapley values from game theory. SHAP estimates feature importance in a consistent manner, which enables an additional validation and/or knowledge extraction. The SHAP method is applicable to any ML model, including regression or ensemble models. Herein, SVM-, RF- and DNN-based predictions were interpreted for exemplary compounds. Structural patterns with large SHAP values were identified and mapped onto 2D graphs of compounds for intuitive visualization. On this basis, model predictions can be structurally interpreted to detect potential sources of bias or confirm model consistency to further validate a model, which is a key factor for ML model acceptance and trust. Overall, the SHAP methodology offers interesting practical opportunities for the further integration of ML models into pharmaceutical research.

Chapter 8

Conclusions

In this thesis, single-target and multi-target ML models have been generated for compound activity prediction. Two prominent modeling problems were tackled: (i) prediction of compound profiling matrices and (ii) identification of strong inhibitors across a panel of kinases. Potential advantages of DL and MT learning over standard ML approaches have been evaluated. Moreover, benchmark settings and the influence of training set conditions on model performance have been investigated. Finally, interpretability of activity predictions has been improved to reduce or eliminate the black box character of complex ML models.

In the first study, distinct prediction strategies and ML methods of varying complexity were benchmarked to model large screening matrices, which contained approximately 110,000 and 143,000 small molecules extensively tested against ∼50 targets. Models based on MT-DNNs and CNNs gave accurate predictions for unrelated tasks but did not outperform other ML methods. Surprisingly, target-based RF models, which are easier to train as well as interpret, yielded successful predictions and detected active compounds for multiple targets.

In the second work, ML strategies were developed to discriminate between highly and weakly potent inhibitors across a set of 103 human kinases. MT learning consistently surpassed ST approaches. MT-DNNs achieved overall the best predictive performance, but advantages over other ML models using ligand data for multiple kinases were only marginal.

Next, SVM classification and ranking models were trained for different activity classes under systematic variation of the number of positive and neg-

ative training compounds. With as few as 50 active compounds for training predictions became accurate. Model performance and stability improved with increasing number of inactives when considering class weights.

Moreover, the effect of compound profiling matrix density on the relative performance of DNN and standard ML methods was systematically analyzed. MT-DNNs slightly improved RF results when average performance across targets was calculated. Nevertheless, for individual targets MT-DNNs only yielded superior performance with very sparse training data.

In the following, SVM and SVR models for compound activity and potency predictions were interpreted and their prioritized model features were analyzed. Surprisingly, fingerprint features contributed very differently to the corresponding SVM and SVR models despite sharing the algorithmic basis.

Finally, an intuitive methodology to interpret activity predictions from complex ML models was proposed. The locally interpretable explanatory method termed SHAP was introduced to rationalize activity predictions of any ML algorithm, regardless of its complexity. The SHAP analysis scheme proposed herein enabled the identification of structural patterns determining activity predictions and their visual access onto compound graphs. Results indicated the high potential of this methodology for rationalizing predictions from complex ML models, including DNNs or ensemble methods.

In conclusion, these studies reflect the potential of ML approaches for mining large compound data sets and modeling bioactivity against individual and multiple targets. Thus, predictive models can guide experimental design as well as increase the enrichment of active compounds in drug discovery projects. Taken together, the results revealed that DL is not a "magic bullet" and only leads to improvements over standard ML methods under certain circumstances. Results provided practical guidelines for ML and prediction of active compounds. Our findings suggest MT-DNNs might be preferred to improve global performance over multiple screens when only sparse matrices are available. Also, MT-DNN might be the method of choice for modeling the activity against a single target with only few known ligands, but when extensive data for other targets are available. In addition, the findings presented herein assign high priority to MT learning schemes when addressing correlated prediction tasks. As also shown, MT modeling of compound-target interactions does not essentially depend on

DNN and can be facilitated with different ML methods. Hence, MT learning in medicinal chemistry has an encouraging perspective for practical applications. Large predictive performance is not sufficient for the usefulness of ML in pharmaceutical research. There is an inherent lack of trust in computational approaches and reluctance to use prediction outcomes without an associated explanation. Accordingly, explainable model decisions provide an additional validation step based on domain knowledge, which might allow the detection of model biases. Furthermore, models can only guide compound modifications if the structural features prioritized by the algorithm are extracted. Therefore, insights on predictive models are required. Herein, the critical issue of model interpretation has been addressed by designing and validating an intuitive analytical methodology. Overall, this dissertation presents contributions to the application and rationalization of ML models for activity predictions and SAR modeling in pharmaceutical research.

Bibliography

[1] Hu, Y.; Bajorath, J. Entering the 'big data' era in medicinal chemistry: molecular promiscuity analysis revisited. *Future Sci. OA* **2017**, *3*.

[2] Bajorath, J. Data analytics and deep learning in medicinal chemistry. *Future Med. Chem.* **2018**, *10*, 1541–1543.

[3] Lo, Y. C.; Rensi, S. E.; Torng, W.; Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **2018**, *23*, 1538–1546.

[4] Hughes, J. P.; Rees, S. S.; Kalindjian, S. B.; Philpott, K. L. Principles of early drug discovery. *Br. J. Pharmacol.* **2011**, *162*, 1239–1249.

[5] Guha, R. On exploring structure-activity relationships. *Methods Mol. Biol.* **2013**, *993*, 81–94.

[6] Winkler, D. A. The role of quantitative structure-activity relationships (QSAR) in biomolecular discovery. *Brief. Bioinform.* **2002**, *3*, 73–86.

[7] Wawer, M.; Lounkine, E.; Wassermann, A. M.; Bajorath, J. Data structures and computational tools for the extraction of SAR information from large compound sets. *Drug Discov. Today* **2010**, *15*, 630–639.

[8] Dudek, A.; Arodz, T.; Galvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb. Chem. High Throughput Screen.* **2006**, *9*, 213–228.

[9] Varnek, A.; Baskin, I. Machine learning methods for property prediction in chemoinformatics: quo vadis? *J. Chem. Inf. Model.* **2012**, *52*, 1413–1437.

[10] Bajorath, J. Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discov.* **2002**, *1*, 882–894.

[11] Lavecchia, A.; Cerchia, C. In silico methods to address polypharmacology: current status, applications and future perspectives. *Drug Discov. Today* **2016**, *21*, 288–298.

[12] Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **2018**, *23*, 1241–1250.

[13] Bhhatarai, B.; Walters, W. P.; Hop, C. E. C. A.; Lanza, G.; Ekins, S. Opportunities and challenges using artificial Intelligence in ADME/Tox. *Nat. Mater.* **2019**, *18*, 418–422.

[14] Scior, T. and Bender, A. and Treandern, G. and Medina-Franco, J. L. and Martinez-Mayorga, K. and Langer, T. and Cuanalo-Contreras, K. and Agrafiotis, D. K., Recognizing Pitfalls in Virtual Screening: a Critical Review. *J. Chem. Inf. Model.* **2012**, *52*, 867–881.

[15] Gawehn, E.; Hiss, J. A.; Schneider, G. Deep learning in drug discovery. *Mol. Inform.* **2016**, *35*, 3–14.

[16] Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **2019**, *18*, 463–477.

[17] Mohs, R. C.; Greig, N. H. Drug discovery and development: role of basic biological research. *Alzheimers Dement.* **2017**, *3*, 651–657.

[18] Sams-Dodd, F. Target-based drug discovery: is something wrong? *Drug Discov. Today* **2005**, *10*, 139–147.

[19] Cohen, P. Protein kinases - The major drug targets of the twenty-first century? *Nat. Rev. Drug Discov.* **2002**, *1*, 309–315.

[20] Lundstrom, K. An overview on GPCRs and drug discovery: structure-based drug design and structural biology on GPCRs. *Methods Mol. Biol.* **2009**, *552*, 51–66.

[21] Inglese, J.; Johnson, R. L.; Simeonov, A.; Xia, M.; Zheng, W.; Austin, C. P.; Auld, D. S. High-throughput screening assays for the identification of chemical probes. *Nat. Chem. Biol.* **2007**, *3*, 466–479.

[22] Mayr, L. M.; Bojanic, D. Novel trends in high-throughput screening. *Curr. Opin. Pharmacol.* **2009**, *9*, 580–8.

[23] DiMasi, J. A.; Grabowski, H. G.; Hansen, R. W. Innovation in the pharmaceutical industry: new estimates of R&D costs. *J. Health Econ.* **2016**, *47*, 20–33.

[24] Xu, J.; Hagler, A. Chemoinformatics and drug discovery. *Molecules* **2002**, *7*, 566–600.

[25] Chen, B.; Butte, A. J. Leveraging big data to transform target selection and drug discovery. *Clin. Pharmacol. Ther.* **2016**, *99*, 285–297.

[26] Jorgensen, W. L. The many roles of computation in drug discovery. *Science* **2004**, *303*, 1813–1818.

[27] Wooller, S. K.; Benstead-Hume, G.; Chen, X.; Ali, Y.; Pearl, F. M. G. Bioinformatics in translational drug discovery. *Biosci. Rep.* **2017**, *37*, BSR20160180.

[28] Brown, F. K. *Annual Reports in Medicinal Chemistry*; Academic Press Inc., 1998; Vol. 33; pp 375–384.

[29] Okuno, Y. In silico drug discovery based on the integration of bioinformatics and chemoinformatics. *Yakugaku Zasshi* **2008**, *128*, 1645–1651.

[30] Krasky, A.; Rohwer, A.; Marhöfer, R. J.; Selzer, P. M. *Antiparasitic and antibacterial drug discovery: from molecular targets to drug candidates*; John Wiley and Sons, 2009; pp 45–57.

[31] Schneider, G. Automating drug discovery. *Nat. Rev. Drug Discov.* **2018**, *17*, 97–113.

[32] Mak, K. K.; Pichika, M. R. Artificial intelligence in drug development: present status and future prospects. *Drug Discov. Today* **2019**, *24*, 773–780.

[33] Hinton, G. Deep learning - a technology with the potential to transform health care. *JAMA* **2018**, *320*, 1101–1102.

[34] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012). 2012.

[35] Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A. R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97.

[36] Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

[37] Rifaioglu, Ahmet Sureyya and Atas, Heval and Martin, Maria Jesus and Cetin-Atalay, Rengul and Atalay, Volkan and Doğan, Tunca, Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Brief. Bioinform.* **2018**, 1–35.

[38] Lavecchia, A. Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug Discov. Today* **2019**, *24*, 2017–2032.

[39] Li, H.; Yap, C. W.; Ung, C. Y.; Xue, Y.; Li, Z. R.; Han, L. Y.; Lin, H. H.; Chen, Y. Z. Machine learning approaches for predicting compounds that interact with therapeutic and ADMET related proteins. *J. Pharm. Sci.* **2007**, *96*, 2838–2860.

[40] Bakheet, T. M.; Doig, A. J. Properties and identification of human protein drug targets. *Bioinformatics* **2009**, *25*, 451–7.

[41] Kraus, V. B. Biomarkers as drug development tools: discovery, validation, qualification and use. *Nat. Rev. Rheumatol.* **2018**, *14*, 354–362.

[42] Jeon, J.; Nim, S.; Teyra, J.; Datti, A.; Wrana, J. L.; Sidhu, S. S.; Moffat, J.; Kim, P. M. A systematic approach to identify novel cancer drug targets using machine learning, inhibitor design and high-throughput screening. *Genome Med.* **2014**, *6*.

[43] Nayal, M.; Honig, B. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins* **2006**, *63*, 892–906.

[44] Iorio, F. et al. A landscape of pharmacogenomic interactions in cancer. *Cell* **2016**, *166*, 740–754.

[45] Rashid, S.; Shah, S.; Bar-Joseph, Z.; Pandya, R. Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. *Bioinformatics* **2019**, in press.

[46] Luecken, M. D.; Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.* **2019**, *15*.

[47] Petegrosso, R.; Li, Z.; Kuang, R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Brief. Bioinform.* **2019**,

[48] Li, B.; Shin, H.; Gulbekyan, G.; Pustovalova, O.; Nikolsky, Y.; Hope, A.; Bessarabova, M.; Schu, M.; Kolpakova-Hart, E.; Merberg, D.; Dorner, A.; Trepicchio, W. L. Development of a drug-response modeling framework to identify cell line derived translational biomarkers that can predict treatment outcome to Erlotinib or Sorafenib. *PLoS ONE* **2015**, *10*, e0130700.

[49] Janowczyk, A.; Madabhushi, A. Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. *J. Pathol. Inform.* **2016**, *7*, 29.

[50] Kraus, O. Z.; Grys, B. T.; Ba, J.; Chong, Y.; Frey, B. J.; Boone, C.; Andrews, B. J. Automated analysis of highcontent microscopy data with deep learning. *Mol. Syst. Biol.* **2017**, *13*, 924.

[51] Cruz-Roa, A.; Gilmore, H.; Basavanhally, A.; Feldman, M.; Ganesan, S.; Shih, N. N. C.; Tomaszewski, J.; González, F. A.; Madabhushi, A. Accurate and reproducible invasive breast cancer detection in whole-slide images: a deep learning approach for quantifying tumor extent. *Sci. Rep.* **2017**, *7*, 46450.

[52] Coley, C. W.; Green, W. H.; Jensen, K. F. Machine learning in computer-aided synthesis planning. *Acc. Chem. Res.* **2018**, *51*, 1281–1289.

[53] Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of organic reaction outcomes using machine learning. *ACS Cent. Sci.* **2017**, *3*, 434–443.

[54] Gao, H.; Struble, T. J.; Coley, C. W.; Wang, Y.; Green, W. H.; Jensen, K. F. Using machine learning to predict suitable conditions for organic reactions. *ACS Cent. Sci.* **2018**, *4*, 1465–1476.

[55] 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*, 268–276.

[56] Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; Zhavoronkov, A. DruGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharm.* **2017**, *14*, 3098–3104.

[57] Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced adversarial neural computer for de novo molecular design. *J. Chem. Inf. Model.* **2018**, *58*, 1194–1204.

[58] Yuan, W.; Jiang, D.; Nambiar, D. K.; Liew, L. P.; Hay, M. P.; Bloomstein, J.; Lu, P.; Turner, B.; Le, Q.-T.; Tibshirani, R.; Khatri, P.; Moloney, M. G.; Koong, A. C. Chemical space mimicry for drug discovery. *J. Chem. Inf. Model.* **2017**, *57*, 875–882.

[59] Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120–131.

[60] Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminformatics* **2017**, *9*.

[61] Li, Q.; Lai, L. Prediction of potential drug targets based on simple sequence properties. *BMC Bioinformatics* **2007**, *8*, 353.

[62] Maltarollo, V. G.; Gertrudes, J. C.; Oliveira, P. R.; Honorio, K. M. Applying machine learning techniques for ADME-Tox prediction: a review. *Expert Opin. Drug Metab. Toxicol.* **2015**, *11*, 259–271.

[63] Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263–274.

[64] Zhou, Y.; Cahya, S.; Combs, S. A.; Nicolaou, C. A.; Wang, J.; Desai, P. V.; Shen, J. Exploring tunable hyperparameters for deep neural networks with industrial ADME data sets. *J. Chem. Inf. Model.* **2019**, *59*, 1005–1016.

[65] Kirchmair, J.; Göller, A. H.; Lang, D.; Kunze, J.; Testa, B.; Wilson, I. D.; Glen, R. C.; Schneider, G. Predicting drug metabolism: experiment and/or computation? *Nat. Rev. Drug Discov.* **2015**, *14*, 387.

[66] Gaulton, A. et al. The ChEMBL database in 2017. *Nucleic Acids Res.* **2017**, *45*, D945–D954.

[67] 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 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2019**, *47*, D1102–D1109.

[68] Eckert, H.; Bajorath, J. Molecular similarity analysis in virtual screening: foundations, limitations and novel approaches. *Drug Discov. Today* **2007**, *12*, 225–33.

[69] Sotriffer, C. Virtual screening : principles, challenges, and practical guidelines. *Curr. Opin. Drug Discov. Devel.* **2009**, *12*, 519.

[70] Lipinski, C. A. Overview of hit to lead: the medicinal chemist's role from HTS retest to lead optimization hand off. *Top. Med. Chem.* **2010**, *5*, 1–24.

[71] Kubinyi, H.; Müller, G. *Chemogenomics in drug discovery: a medicinal chemistry perspective*; Wiley-VCH, 2004; p 463.

[72] Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **2004**, *432*, 862–865.

[73] Leach, A. R.; Gillet, V. J. *An introduction to chemoinformatics*; Springer Netherlands, 2007; p 255.

[74] de Ruyck, J.; Brysbaert, G.; Blossey, R.; Lensink, M. F. Molecular docking as a popular tool in drug design, an in silico travel. *Adv. Appl. Bioinform. Chem.* **2016**, *9*.

[75] Pagadala, N. S.; Syed, K.; Tuszynski, J. Software for molecular docking: a review. *Biophys. Rev.* **2017**, *9*, 91–102.

[76] Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat. Rev. Drug Discov.* **2004**, *3*, 935–949.

[77] Shen, C.; Ding, J.; Wang, Z.; Cao, D.; Ding, X.; Hou, T. From machine learning to deep learning: advances in scoring functions for protein-ligand docking. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2019**, e1429.

[78] Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.* **2017**, *57*, 942–957.

[79] Willett, P.; Barnard, J. M.; Downs, G. M. Chemical similarity searching. *J. Chem. Inf. Model.* **1998**, *38*, 983–996.

[80] Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug Discov. Today* **2015**, *20*, 318–31.

[81] Hu, Y.; Stumpfe, D.; Bajorath, J. Recent advances in scaffold hopping. *J. Med. Chem.* **2017**, *60*, 1238–1246.

[82] Saeh, J. C.; Lyne, P. D.; Takasaki, B. K.; Cosgrove, D. A. Lead hopping using SVM and 3D pharmacophore fingerprints. *J. Chem. Inf. Model.* **2005**, *45*, 1122–33.

[83] Laufkötter, O.; Sturm, N.; Bajorath, J.; Chen, H.; Engkvist, O. Combining structural and bioactivity-based fingerprints improves prediction performance and scaffold hopping capability. *J. Cheminformatics* **2019**, *11*.

[84] Doddareddy, M. R.; Klaasse, E. C.; Shagufta,; Ijzerman, A. P.; Bender, A. Prospective validation of a comprehensive in silico hERG model and its applications to commercial compound and drug databases. *ChemMedChem* **2010**, *5*, 716–729.

[85] Franke, L.; Byvatov, E.; Werz, O.; Steinhilber, D.; Schneider, P.; Schneider, G. Extraction and visualization of potential pharmacophore points using support vector machines: application to ligand-based virtual screening for COX-2 inhibitors. *J. Med. Chem.* **2005**, *48*, 6997–7004.

[86] Ripphausen, P.; Nisius, B.; Peltason, L.; Bajorath, J. Quo vadis, virtual screening? A comprehensive survey of prospective applications. *J. Med. Chem.* **2010**, *53*, 8461–8467.

[87] Carpenter, K. A.; Huang, X. Machine learning-based virtual screening and its applications to Alzheimer's drug discovery: a review. *Curr. Pharm.* **2018**, *24*, 3347–3358.

[88] Tang, H.; Wang, X. S.; Huang, X.-P.; Roth, B. L.; Butler, K. V.; Kozikowski, A. P.; Jung, M.; Tropsha, A. Novel inhibitors of human histone deacetylase (HDAC) identified by QSAR modeling of known inhibitors, virtual screening, and experimental validation. *J. Chem. Inf. Model.* **2009**, *49*, 461–76.

[89] Yu, M.; Gu, Q.; Xu, J. Discovering new PI3Kα inhibitors with a strategy of combining ligand-based and structure-based virtual screening. *J. Comput. Aided Mol. Des.* **2018**, *32*, 347–361.

[90] Fujiwara, Y.; Yamashita, Y.; Osoda, T.; Asogawa, M.; Fukushima, C.; Asao, M.; Shimadzu, H.; Nakao, K.; Shimizu, R. Virtual screening system for finding structurally diverse hits by active learning. *J. Chem. Inf. Model.* **2008**, *48*, 930–940.

[91] Reker, D.; Schneider, P.; Schneider, G.; Brown, J. B. Active learning for computational chemogenomics. *Future Med. Chem.* **2017**, *9*, 381–402.

[92] Reker, D.; Schneider, G. Active-learning strategies in computer-assisted drug discovery. *Drug Discov. Today* **2015**, *20*, 458–465.

[93] Bolognesi, M. L.; Cavalli, A. Multitarget drug discovery and polypharmacology. *ChemMedChem* **2016**, *11*, 1190–2.

[94] Maggiora, G.; Gokhale, V. Non-specificity of drug-target interactions - Consequences for drug discovery. Frontiers in Molecular Design and Chemical Information Science (ACS Symposium Series). 2016; pp 91–142.

[95] Bottegoni, G.; Favia, A. D.; Recanatini, M.; Cavalli, A. The role of fragment-based and computational methods in polypharmacology. *Drug Discov. Today* **2012**, *17*, 23–34.

[96] Boran, A. D. W.; Iyengar, R. Systems approaches to polypharmacology and drug discovery. *Curr. Opin. Drug Discov. Devel.* **2010**, *13*, 297–309.

[97] Schuffenhauer, A.; Floersheim, P.; Acklin, P.; Jacoby, E. Similarity metrics for ligands reflecting the similarity of the target proteins. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 391–405.

[98] Jacob, L.; Vert, J.-P. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* **2008**, *24*, 2149–56.

[99] Cheng, F.; Zhou, Y.; Li, J.; Li, W.; Liu, G.; Tang, Y. Prediction of chemical-protein interactions: multitarget-QSAR versus computational chemogenomic methods. *Mol. Biosyst.* **2012**, *8*, 2373–2384.

[100] Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is multitask deep learning practical for pharma? *J. Chem. Inf. Model.* **2017**, *57*, 13.

[101] Geppert, H.; Humrich, J.; Stumpfe, D.; Gärtner, T.; Bajorath, J. Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors. *J. Chem. Inf. Model.* **2009**, *49*, 767–779.

[102] Wassermann, A. M.; Geppert, H.; Bajorath, J. Ligand prediction for orphan targets using support vector machines and various target-ligand kernels is dominated by nearest neighbor effects. *J. Chem. Inf. Model.* **2009**, *49*, 2155–67.

[103] Van Westen, G. J. P.; Wegner, J. K.; Ijzerman, A. P.; Van Vlijmen, H. W. T.; Bender, A. Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets. *MedChemComm* **2011**, *2*, 16–30.

[104] Ekins, S. The next era: deep learning in pharmaceutical research. *Pharm. Res.* **2016**, *33*, 2594–2603.

[105] Gilvary, C.; Madhukar, N.; Elkhader, J.; Elemento, O. The missing pieces of artificial intelligence in medicine. *Trends Pharmacol. Sci.* **2019**, *40*, 555–564.

[106] Heaven, D. Deep trouble for deep learning. *Nature* **2019**, *574*, 163–166.

[107] Mitchell, J. B. O. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*.

[108] Cherkasov, A. et al. QSAR modeling: where have you been? where are you going to? *J. Med. Chem.* **2014**, *57*, 4977–5010.

[109] Bajorath, J. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 233–245.

[110] Polanski, J.; Gasteiger, J. *Handbook of Computational Chemistry*; Springer International Publishing, 2017; pp 1997–2039.

[111] Bonchev, D.; Rouvray, D. H. *Chemical graph theory: introduction and fundamentals*; Abacus Press, 1991; p 288.

[112] Anderson, E; Veith, GD; Weininger, D. *SMILES: a line notation and computerized interpreter for chemical structures (EPA/600/M-87/021)*; 1987.

[113] Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC international chemical identifier. *J. Cheminformatics* **2015**, *7*.

[114] Accelrys, MACCS keys. MDL Information Systems, Inc. 2011.

[115] Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

[116] Bjerrum, E. J.; Sattarov, B. SMILES enumeration as data augmentation for neural network modeling of molecules. *Biomolecules* **2018**, *8*.

[117] Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. Advances in Neural Information Processing Systems 28 (NIPS 2015). 2015.

[118] Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* **2016**, *30*, 595–608.

[119] Johnson, A. M.; Maggiora, G. M. *Concepts and applications of molecular similarity*; John Willey & Sons: New York, 1990.

[120] Kubinyi, H. *3D QSAR in Drug Design*; Kluwer Academic Publishers: Dordrecht, 1998; pp 225–252.

[121] Bender, A.; Glen, R. C. Molecular similarity: a key technique in molecular informatics. *Org. Biomol. Chem.* **2004**, *2*, 3204–18.

[122] Maggiora, G.; Vogt, M.; Stumpfe, D.; Bajorath, J. Molecular similarity in medicinal chemistry. *J. Med. Chem.* **2014**, *57*, 3186–3204.

[123] Hussain, J.; Rea, C. Computationally efficient algorithm to identify matched molecular pairs (MMPs) in large data sets. *J. Chem. Inf. Model.* **2010**, *50*, 339–48.

[124] Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP - retrosynthetic combinatorial analysis procedure: a powerful new technique

for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511–522.

[125] Stumpfe, D.; Dimova, D.; Bajorath, J. Computational method for the systematic identification of analog series and key compounds representing series and their biological activity profiles. *J. Med. Chem.* **2016**, *59*, 7667–7676.

[126] Dimova, D.; Stumpfe, D.; Hu, Y.; Bajorath, J. Analog series-based scaffolds: computational design and exploration of a new type of molecular scaffolds for medicinal chemistry. *Future Sci. OA* **2016**, *2*, FSO149.

[127] Willett, P. Similarity-based virtual screening using 2D fingerprints. *Drug Discov. Today* **2006**, *11*, 1046–53.

[128] Stumpfe, D.; Bajorath, J. Similarity searching. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2011**, *1*, 260–282.

[129] Heikamp, K.; Bajorath, J. Large-scale similarity search profiling of ChEMBL compound data sets. *J. Chem. Inf. Model.* **2011**, *51*, 1831–1839.

[130] Willett, P. Combination of similarity rankings using data fusion. *J. Chem. Inf. Model.* **2013**, *53*, 1–10.

[131] Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *J. Chem. Inf. Model.* **2006**, *46*, 462–470.

[132] Gardiner, E. J.; Gillet, V. J.; Haranczyk, M.; Hert, J.; Holliday, J. D.; Malim, N.; Patel, Y.; Willett, P. Turbo similarity searching: effect of fingerprint and dataset on virtual-screening performance. *Stat. Anal. Data Min.* **2009**, *2*, 103–114.

[133] Xue, L.; Stahura, F. L.; Godden, J. W.; Bajorath, J. Fingerprint scaling increases the probability of identifying molecules with similar activity in virtual screening calculations. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 746–753.

[134] Xue, L.; Godden, J. W.; Stahura, F. L.; Bajorath, J. Profile scaling increases the similarity search performance of molecular fingerprints containing numerical descriptors and structural keys. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1218–1225.

[135] Vogt, M.; Bajorath, J. Introduction of the conditional correlated Bernoulli model of similarity value distributions and its application to the prospective prediction of fingerprint search performance. *J. Chem. Inf. Model.* **2011**, *51*, 2496–2506.

[136] Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning*, 2nd ed.; Springer, 2009.

[137] Krstajic, D.; Buturovic, L. J.; Leahy, D. E.; Thomas, S. Cross-validation pitfalls when selecting and assessing regression and classification models. *J. Cheminformatics* **2014**, *6*, 1–15.

[138] Breiman, L.; Spector, P. Submodel selection and evaluation in regression. The X-random case. *Int. Stat. Rev.* **1992**, *60*, 291.

[139] Bishop, C. M. *Pattern recognition and machine learning*; Springer, 2006; p 738.

[140] Vogt, M.; Bajorath, J. Bayesian screening for active compounds in high-dimensional chemical spaces combining property descriptors and molecular fingerprints. *Chem. Biol. Drug Des.* **2008**, *71*, 8–14.

[141] Breiman, L. Random forests. *Statistics* **2001**, *45*, 1–33.

[142] Breiman, Leo; Friedman, Jerome; Stone, Charles J.; Olshen, R. *Classification and regression trees*, 1st ed.; CRC Press: Taylor & Francis Group, 1984.

[143] Breiman, L. Bagging predictors. *Machine Learning* **1996**, *24*, 123–140.

[144] Cortes, C. and Vapnik, Vladimir N., Support-vector networks. *Machine Learning* **1995**, *20*, 273–297.

[145] Vapnik, V. N. *The nature of statistical learning theory*; Springer New York, 1995.

[146] Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Disc.* **1998**, *2*, 121–167.

[147] Jorissen, R. N.; Gilson, M. K. Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.* **2005**, *45*, 549–561.

[148] Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. Advances in Neural Information Processing Systems 9 (NIPS 1996). 1996.

[149] Smola, Alex and Schölkopf, B., A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222.

[150] Alpaydin, E. *Introduction to machine learning*, 2nd ed.; The MIT Press: Cambridge, Massachusetts, USA, 2010.

[151] Hofmann, T.; Schölkopf, B.; Smola, A. J. Kernel methods in machine learning. *The Annals of Statistics* **2008**, *36*, 1171–1220.

[152] Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Networks* **2005**, *18*, 1093–1110.

[153] Nielsen, M. A. *Neural networks and deep learning*; Determination Press, 2015.

[154] Goodfellow, Ian J; Bengio, Y.; Courville, A., *Deep learning*; The MIT Press: Cambridge, Massachusetts, USA, 2016.

[155] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536.

[156] Kiefer, J.; Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* **1952**, *23*, 462–466.

[157] Altae-Tran, H.; Ramsundar, B.; Pappu, A. S.; Pande, V. Low data drug discovery with one-shot learning. *ACS Cent. Sci.* **2017**, *3*, 283–293.

[158] Caruana, R.; Pratt, L.; Thrun, S. Multitask learning. **1997**, *28*, 41–75.

[159] Brown, J. B.; Okuno, Y.; Marcou, G.; Varnek, A.; Horvath, D. Computational chemogenomics: is it more than inductive transfer? *J. Comput. Aided Mol. Des.* **2014**, *28*, 597–618.

[160] Lapinsh, M.; Prusis, P.; Lundstedt, T.; Wikberg, J. E. S. Proteochemometrics modeling of the interaction of amine G-protein coupled receptors with a diverse set of ligands. *Mol. Pharm.* **2002**, *61*, 1465–75.

[161] Cortés-Ciriano, I.; Ain, Q. U.; Subramanian, V.; Lenselink, E. B.; Méndez-Lucio, O.; Ijzerman, A. P.; Wohlfahrt, G.; Prusis, P.; Malliavin, T.; Van Westen, G. J. P.; Bender, A. Polypharmacology modelling using proteochemometrics (PCM): recent methodological developments, applications to target families, and future prospects. *MedChemComm* **2015**, *6*, 24–50.

[162] Stanton, D. T. On the physical interpretation of QSAR models. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1423–1433.

[163] Guha, R. On the interpretation and interpretability of quantitative structure-activity relationship models. *J. Comput. Aided Mol. Des.* **2008**, *22*, 857–871.

[164] Polishchuk, P. Interpretation of quantitative structureactivity relationship models: past, present, and future. *J. Chem. Inf. Model.* **2017**, *57*, 2618–2639.

[165] Shoombuatong, W.; Prathipati, P.; Owasirikul, W.; Worachartcheewan, A.; Simeon, S.; Anuwongcharoen, N.; Wikberg, J. E. S.; Nantasenamat, C. *Advances in QSAR modeling*; Springer, 2017; pp 3–55.

Additional publications

Miljković, F.; Rodríguez-Pérez, R.; Bajorath, J. Machine Learning Models for Accurate Prediction of Kinase Inhibitors with Different Binding Modes. *J. Med. Chem.* **2019**, doi: 10.1021/acs.jmedchem.9b00867.

Rodríguez-Pérez, R.; Bajorath, J. Support Vector Machines in Pharmaceutical Research, In *Artificial Intelligence and Machine Learning in Drug Development*, Matsson, P., Bergström, C., Eds.; AAPS Advances in Pharmaceutical Sciences, Springer Nature Switzerland AG, in press.