Olier, I, Sadawi, N, Bickerton, GR, Vanschoren, J, Grosan, C, Soldatova, L and King, RD

Meta-QSAR: a large-scale application of meta-learning to drug design and discovery

Mach Learn (2018) 107:285–311
https://doi.org/10.1007/s10994-017-5685-x

Meta-QSAR: a large-scale application of meta-learning to drug design and discovery

Ivan Olier[1,2] · Noureddin Sadawi[3,4] · G. Richard Bickerton[5,6] · Joaquin Vanschoren[7] · Crina Grosan[4,8] · Larisa Soldatova[4,9] · Ross D. King[2]

Received: 9 May 2016 / Accepted: 4 October 2017 / Published online: 22 December 2017

Abstract We investigate the learning of quantitative structure activity relationships (QSARs) as a case-study of meta-learning. This application area is of the highest societal importance, as it is a key step in the development of new medicines. The standard QSAR learning problem

Editors: Pavel Brazdil and Christophe Giraud-Carrier.

✉ Crina Grosan
crina.grosan@brunel.ac.uk

Ivan Olier
i.olier@mmu.ac.uk

Noureddin Sadawi
n.sadawi@imperial.ac.uk

G. Richard Bickerton
g.r.bickerton@dundee.ac.uk; rbickerton@exscientia.co.uk

Joaquin Vanschoren
j.vanschoren@tue.nl

Larisa Soldatova
larisa.soldatova@brunel.ac.uk

Ross D. King
r.king@manchester.ac.uk

1 Manchester Metropolitan University, Manchester, UK

2 University of Manchester, Manchester, UK

3 Imperial College London, London, UK

4 Brunel University London, London, UK

5 Dundee University, Dundee, UK

6 Exscientia Ltd, Dundee, UK

7 Eindhoven University of Technology, Eindhoven, The Netherlands

8 Babes-Bolyai University, Cluj-Napoca, Romania

9 Goldsmiths, University of London, London, UK

is: given a target (usually a protein) and a set of chemical compounds (small molecules) with associated bioactivities (e.g. inhibition of the target), learn a predictive mapping from molecular representation to activity. Although almost every type of machine learning method has been applied to QSAR learning there is no agreed single best way of learning QSARs, and therefore the problem area is well-suited to meta-learning. We first carried out the most comprehensive ever comparison of machine learning methods for QSAR learning: 18 regression methods, 3 molecular representations, applied to more than 2700 QSAR problems. (These results have been made publicly available on OpenML and represent a valuable resource for testing novel meta-learning methods.) We then investigated the utility of algorithm selection for QSAR problems. We found that this meta-learning approach outperformed the best individual QSAR learning method (random forests using a molecular fingerprint representation) by up to 13%, on average. We conclude that meta-learning outperforms base-learning methods for QSAR learning, and as this investigation is one of the most extensive ever comparisons of base and meta-learning methods ever made, it provides evidence for the general effectiveness of meta-learning over base-learning.

Keywords Meta-learning · Algorithm selection · Drug discovery · QSAR

1 Introduction

The standard approach to predicting how active a chemical compound will be against a given target (usually a protein that needs to be inhibited) in the development of new medicines is to use machine learning models. Currently, there is no agreed single best learning algorithm to do this. In this paper we investigate the utility of meta-learning to address this problem. We aim to discover and exploit relationships between machine learning algorithms, measurable properties of the input data, and the empirical performance of learning algorithms, to infer the best models to predict the activity of chemical compounds on a given target.

1.1 Quantitative structure activity relationship (QSAR) learning

Drug development is one of the most important applications of science, as it is an essential step in the treatment of almost all diseases. Developing a new drug is however slow and expensive. The average cost to bring a new drug to market is >2.5 billion US dollars (DiMasi et al. 2015), which means that tropical diseases such as malaria, schistosomiasis, Chagas' disease, etc., which kill millions of people and infect hundreds of millions of others are 'neglected' (Ioset and Chang 2011; Leslie and Inouye 2011) and that 'orphan' diseases (i.e. those with few sufferers) remain untreatable (Braun et al. 2010). More generally, the pharmaceutical industry is struggling to cope with spiralling drug discovery and development costs (Pammolli et al. 2011). Drug development is also slow, generally taking more than 10 years. This means that there is strong pressure to speed up development, both to save lives and reduce costs. A successful drug can earn billions of dollars a year, and as patent protection is time-limited, even one extra week of patent protection can be of great financial significance.

A key step in drug development is learning Quantitative Structure Activity Relationships (QSARs) (Martin 2010; Cherkasov et al. 2014; Cumming et al. 2013). These are functions that predict a compound's bioactivity from its structure. The standard QSAR learning problem is: given a target (usually a protein) and a set of chemical compounds (small molecules) with associated bioactivities (e.g. inhibiting the target), learn a predictive mapping from molecular representation to activity.

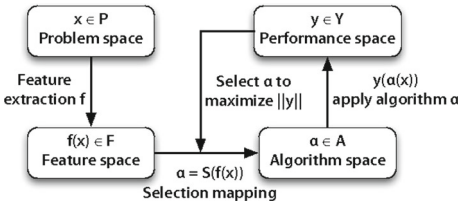


Fig. 1 Rice's framework for algorithm selection. Adapted from Rice (1976) and Smith-Miles (2008)

Although almost every form of statistical and machine learning method has been applied to learning QSARs, there is no agreed single best way of learning QSARs. Therefore an important motivation for this work is to better understand the performance characteristics of the main (baseline) machine learning methods currently used in QSAR learning. This knowledge will feed into a better understanding of the performance characteristics of these algorithms, and will enable QSAR practitioners to improve there predictions.

The central motivation for this work is to better understand meta-learning through a case-study in the very important real-world application area of QSAR learning. This application area is an excellent test-bed for the development of meta-learning methodologies. The importance of the subject area means that there are now thousands of publicly available QSAR datasets, all with the same basic structure. Few machine learning application areas have so many datasets—enabling statistical confidence in meta-learning results. In investigating meta-learning we have focused on algorithm selection as this is the simplest form of meta-learning, and its use fits in with our desire to better understand the baseline-learning methods.

A final motivation for the work is to improve the predictive performance of QSAR learning through use of meta-learning. Our hope is that improved predictive performance will feed into faster and cheaper drug development.

To enable others to build on our base-learning and meta-learning work we have placed all our results in OpenML.

1.2 Meta-learning: algorithm selection

Meta-learning has been used extensively to select the most appropriate learning algorithm on a given dataset. In this section, we first sketch a general framework for algorithm selection, and then provide an overview of prior approaches and the state-of-the-art in selecting algorithms using meta-learning.

1.2.1 Algorithm selection framework

The algorithm selection framework contains four main components: *First*, we construct the problem space P, in our case the space of all QSAR datasets. Each dataset expresses the properties and activity of a limited set of molecular compounds (drugs) on a specific target protein. In this paper, we consider 8292 QSAR datasets, described in more detail in Sect. 2.2. *Second*, we describe each QSAR dataset in P with a set of measurable characteristics (meta-features), yielding the feature space F. In this paper we include two types of meta-features: those that describe the QSAR data itself (e.g. the number of data points), and those that describe properties of the target protein (e.g. hydrophobicity). We expect that these properties will affect the interplay of different QSAR features, and hence the choice of learning algorithm. The full set of meta-features used in this paper is described in Sect. 3.

Third, the algorithm space A is created by the set of all candidate base-level learning algorithms, in our case a set of 18 regression algorithms combined with several preprocessing steps. These are described in Sect. 2.1.

Finally, the performance space Y represents the empirically measured performance, e.g. root mean squared error (RMSE) (Witten and Frank 2005) of each algorithm A on each of the QSAR datasets in P.

In the current state-of-the-art, there exists a wide variety of algorithm selection algorithms. If only a single algorithm should be run, we can train a classification model that makes exactly that prediction (Pfahringer et al. 2000; Guerri and Milano 2012). We can also use a regression algorithm to predict the performance of each algorithm (Xu et al. 2008), build a ranking of promising algorithms (Leite et al. 2012), or use cost-sensitive techniques which allow us to optimize the loss we really care about in the end (Bischl et al. 2012; Xu et al. 2012).

Our task is: for any given QSAR problem $x \in P$, select the best combination of QSAR and molecular representation $a \in A$ that maximizes a predefined performance measure $y \in Y$. In this paper, we investigate two meta-learning approaches: (1) classification problem: the aim is to learn a model that captures the relationship between the properties of the QSAR datasets, or meta-data, and the performance of the regression algorithms. This model can then be used to predict the most suitable algorithm for a new dataset. (2) ranking problem: the aim is to fit a model that ranks the QSAR combinations by their predicted performances.

1.2.2 Previous work on algorithm selection using meta-learning

Meta-features In the meta-learning literature much effort has been devoted to the development of meta-features that effectively describe the characteristics of the data. These should have discriminative power, meaning that they should be able to distinguish between base-learners in terms of their performance, and have a low computational complexity—preferably lower than $O(n \log n)$ (Pfahringer et al. 2000). Meta-features are typically categorised as one of the following: simple (e.g. number of data points, number of features), statistical (e.g. mean standard deviation of attributes, mean kurtosis of attributes, mean skewness of attributes), or information theoretic (e.g. mean entropy of the features, noise–signal ratio). See Bickel et al. (2008), Kalousis (2002) and Vanschoren (2010) for an extensive description of meta-features. A subset of these may be used for regression, and some measures are specifically defined for regression targets (Soares et al. 2004). Other meta-features can be trivially adapted to the regression data. First, landmarking (Pfahringer et al. 2000) works by training and evaluating sets of simple, fast algorithms on the datasets (e.g. a decision stump instead of a full decision tree), and using their performance (e.g. RMSE) as meta-features for the dataset. An analysis of landmarkers for regression problems can be found in Ler et al. (2005).

Another approach is to use model-based characteristics (Peng et al. 2002), obtained by building fast, interpretable models, e.g. decision trees, and then extracting properties of those models, such as the width, the depth and the number of leaves in the tree, and statistical properties (min, max, mean, stdev) of the distribution of nodes in each level of the tree, branch lengths, or occurrences of features in the splitting tests in the nodes. Recent research on finding interesting ways to measure data characteristics includes instance-level complexity (Smith et al. 2014a), measures for unsupervised learning (Lee and Giraud-Carrier 2011), and discretized meta-features (Lee and Giraud-Carrier 2008).

Meta-learning has also been successfully applied in stream mining (van Rijn et al. 2015b, 2014) and time series analysis (Prudêncio and Ludermir 2004), each time requiring novel sets of meta-features.

Selecting algorithms In meta-learning, algorithm selection is traditionally seen as a learning problem: train a meta-learner that predicts the best algorithm(s) given a set of meta-features describing the data. In the setting of selecting a best single algorithm, experiments on artificial datasets showed that there is no single best meta-learner, but that decision tree-like algorithms (e.g. C5.0boost) seem to have an edge, especially when used in combination with landmarkers (Bensusan and Giraud-Carrier 2000; Pfahringer et al. 2000). Further experiments performed on real-world data corroborated these results, although they also show that most meta-learners are very sensitive to the exact combination of meta-features used (Köpf et al. 2000).

In the setting of recommending a subset of algorithms it was shown that, when using statistical and information-theoretical meta-features, boosted decision trees obtained best results (Kalousis 2002; Kalousis and Hilario 2001). Relational case-based reasoning has also been successfully applied (Lindner and Studer 1999; Hilario and Kalousis 2001), which allows to include algorithm properties independent of the dataset and histogram representations of dataset attribute properties.

Most relevant for this paper is the work by Amasyali and Ersoy (2009), which uses around 200 meta-features to select the best regression algorithm for a range of artificial, benchmarking, and drug discovery datasets. The reported correlations between meta-features and algorithm performances were typically above 0.9 on artificial and benchmarking datasets, but much worse (below 0.8) on the drug discovery datasets. Feature selection was found to be important to improve meta-learning performance.

Ranking algorithms Another approach is to build a ranking of algorithms, listing which algorithms to try first. Several techniques use k-nearest neighbors (Brazdil et al. 2003; dos Santos et al. 2004), and compute the average rank (or success rate ratio's or significant wins) over all similar prior datasets (Soares and Brazdil 2000; Brazdil and Soares 2000). Other approaches directly estimate the performances of all algorithms (Bensusan and Kalousis 2001), or use predictive clustering trees (Todorovski et al. 2002).

Better results were obtained by *subsampling landmarkers*, i.e. running all candidate algorithms on several small samples of the new data (Fürnkranz and Petrak 2001). Meta-learning on data samples (MDS) (Leite and Brazdil 2005, 2007) builds on this idea by first determining the complete learning curves of a number of learning algorithms on several different datasets. Then, for a new dataset, progressive subsampling is done up to a certain point, creating a partial learning curve, which is then matched to the nearest complete learning curve for each algorithm in order to predict their final performances on the entire new dataset.

Another approach is to sequentially evaluate a few algorithms on the (complete) new dataset and learn from these results. *Active testing* (Leite et al. 2012) proceeds in a tournament-style fashion: in each round it selects and tests the algorithm that is most likely to outperform the current best algorithm, based on a history of prior duels between both algorithms on similar datasets. Each new test will contribute information to a better estimate of dataset similarity, and thus help to better predict which algorithms are most promising on the new dataset. Large-scale experiments show that active testing outperforms previous approaches, and yields an algorithm whose performance is very close to the optimum, after relatively few tests. More recent work aims to speed up active testing by combining it with learning curves (van Rijn et al. 2015a), so that candidate algorithms only need to be trained on a smaller sample of the data. It also uses a multi-objective criterion called AR3 (Abdulrahman and Brazdil 2014) that trades off runtime and accuracy so that fast but reasonably accurate candidates are evaluated first. Experimental results show that this method converges extremely fast to an acceptable solution.

Finally, algorithms can also be ranked using collaborative filtering (Bardenet et al. 2013; Misir and Sebag 2013; Smith et al. 2014b). In this approach, previous algorithm evaluations are used as 'ratings' for a given dataset. For a new dataset, algorithms which would likely perform well (give a high rating) are selected based on collaborative filtering models (e.g. using matrix decompositions).

Model-based optimization Model-based optimization (Hutter et al. 2011) aims to select the best algorithm and/or best hyperparameter settings for a given dataset by sequentially evaluating them on the full dataset. It learns from prior experiments by building a surrogate model that predicts which algorithms and parameters are likely to perform well. An approach that has proven to work well in practice is Bayesian Optimization (Brochu et al. 2010), which builds a surrogate model (e.g. using Gaussian Processes or Random Forests) to predict the expected performance of all candidate configurations, as well as the uncertainty of that prediction. In order to select the next candidate to evaluate, an acquisition function is used that trades off exploitation (choosing candidates in regions known to perform well) versus exploration (trying candidates in relatively unexplored regions). Bayesian Optimization is used in Auto-WEKA (Thornton et al. 2013) and Auto-sklearn (Feurer et al. 2015), which search for the optimal algorithms and hyperparameters across the *WEKA* (Hall et al. 2009) and *scikit-learn* (Pedregosa et al. 2011) environments, respectively. Given that this technique is computationally very expensive, recent research has tried to include meta-learning to find a good solution faster. One approach is to find a good set of initial candidate configurations by using meta-learning (Feurer et al. 2015): based on meta-features, one can find the most similar datasets and use the optimal algorithms and parameter settings for these datasets as the initial candidates to evaluate. In effect, this provides a 'warm start' which yields better results faster.

1.3 Meta-QSAR learning

Almost every form of statistical and machine learning method has been applied to learning QSARs: linear regression, decision trees, neural networks, nearest-neighbour methods, support vector machines, Bayesian networks, relational learning, etc. These methods differ mainly in the a *a priori* assumptions they make about the learning task. We focus on regression algorithms as this is how QSAR problems are normally cast.

For Meta-QSAR learning the input data are datasets of compound activity (one for each target protein), different representations of the structures of the compounds, and we aim to learn to predict how well different learning algorithms perform, and to exploit these predictions to improve QSAR predictions. We expect meta-learning to be successful for QSAR because although all the datasets have the same overall structure, they differ in the numbers of data points (tested chemical compounds), in the range and occurrence of features (compound descriptors), and in the type of chemical/biochemical mechanism that causes the bioactivity. These differences indicate that different machine learning methods are to be used for different kinds of QSAR data.

We first applied meta-learning to predict the machine learning algorithm that is expected to perform best on a given QSAR dataset. This is known as the algorithm selection problem, and can be expressed formally using Rice's framework for algorithm selection (Rice 1976) as illustrated in Fig. 1. We then applied multi-task learning to first test whether it can improve on standard QSAR learning through the exploitation of evolutionary related targets, and whether multi-task learning can further be improved by incorporating the evolutionary distance of targets.

1.4 Paper outline

The remainder of this paper is organized as follows. In Sect. 2, we report our baseline experiments investigating the effectiveness of a large number of regression algorithms on thousands of QSAR datasets, using different data representations. In Sect. 3 we describe a novel set of QSAR-specific meta-features to inform our meta-learning approach. In Sect. 4 we investigate the utility of meta-learning for selecting the best algorithm for learning QSARs. Finally, Sect. 5 presents a discussion of our results and future work.

2 Baseline QSAR learning

We first performed experiments with a set of baseline regression algorithms to investigate their effectiveness on QSAR problems. Learning a QSAR model consists of fitting a regression model to a dataset which has as instances the chemical compounds, as input variables the chemical compound descriptors, and as numeric response variable (output) the associated bioactivities.

2.1 Baseline QSAR learning algorithms

For our baseline QSAR methods we selected 18 regression algorithms, including linear regression, support vector machines, artificial neural networks, regression trees, and random forests. Table 1 lists all the algorithms used and their respective parameter settings. Within the scope of this study, we do not optimize the parameter settings on every dataset, but instead chose values that are likely to perform well on most QSAR datasets. This list includes the most commonly used QSAR methods in the literature.

With the exception of one of the neural networks implementations, for which we used the H2O R package,[1] all of the algorithms were implemented using the MLR R package for machine learning.[2]

2.2 Baseline QSAR datasets

For many years, QSAR research was held back by a lack of openly available datasets. This situation has been transformed by a number of developments. The most important of these is the open availability of the ChEMBL database,[3] a medicinal chemistry database managed by the European Bioinformatics Institute (EBI). It is abstracted and curated from the scientific literature, and covers a significant fraction of the medicinal chemistry corpus. The data consist of information on the drug targets (mainly proteins from a broad set of target families, e.g. kinases), the structures of the tested compounds (from which different chemoinformatic representations may be calculated), and the bioactivities of the compounds on their targets, such as binding constants, pharmacology, and toxicity. The key advantages of using ChEMBL for Meta-QSAR are: (a) it covers a very large number of targets, (b) the diversity of the chemical space investigated, and (c) the high quality of the interaction data. Its main weakness is that for any single target, interaction data on only a relatively small number of compounds are given.

[1] https://cran.r-project.org/web/packages/h2o/index.html.

[2] https://cran.r-project.org/web/packages/mlr/index.html.

[3] https://www.ebi.ac.uk/chembl/.

Table 1 List of baseline QSAR algorithms

Short name	Name	Parameter settings
ctree	Conditional trees	min_split = 20, min_bucket = 7
rtree	Regression trees	min_split = 20, min_bucket = 7
cforest	Random forest (with conditional trees)	n_trees = 500, min_split = 20, min_bucket = 7
rforest	Random forest	n_trees = 500, min_split = 20, min_bucket = 7
gbm	Generalized boosted regression	n_trees = 100, depth = 1, CV = no, min_obs_node = 10
fnn	k-Nearest neighbor	k = 1
earth	Adaptive regression splines (earth)	(As default)
glmnet	Regularized GLM	(As default)
ridge	Penalized ridge regression	(As default)
lm	Multiple linear regression	(As default)
pcr	Principal component regression	(As default)
plsr	Partial least squares	(As default)
rsm	Response surface regression	(As default)
rvm	Relevance vector machine	Kernel = RBF, nu = 0.2, epsilon = 0.1
ksvm	Support vector machines	Kernel = RBF, nu = 0.2, epsilon = 0.1
ksvmfp	Support vector machines with Tanimoto kernel	Kernel = Tanimoto
nnet	Neural networks	size = 3
nneth2o	Neural networks using H2O library	layers = 2, size layer 1 = 0.333* n_inputs, layer 2 = 0.667*n_inputs

n_trees number of trees; *min_split* minimum node size allowed for splitting; *min_bucket* minimum size of the bucket. *k* number of neighbours; *depth* search depth; *CV* cross-validation; *min_obs_node* minimum number of observations per node; *RBF* radial basis function with nu (spread) and epsilon (scale) parameters; *size* number of neurons in the hidden layer; *n_inputs* length of the input vector

Table 2 Names of the generated dataset representations

	Basic set of descriptors (43)	All descriptors (1447)	FCFP4 fingerprint (1024)
Original dataset	basicmolprop (not used)	allmolprop (not used)	fpFCFP4
Missing value imputation	basicmolprop.miss	allmolprop.miss	(No missing values)

We extracted 2764 targets from ChEMBL with a diverse number of chemical compounds, ranging from 10 to about 6000, each target resulting in a dataset with as many examples as compounds. The target (output) variable contains the associated bioactivities. Bioactivity data were selected on the basis that the target type is a protein, thereby excluding other potential targets such as cell-based and in vivo assays, and the activity type is from a defined list of potency/affinity endpoints (IC50, EC50, Ki, Kd and their equivalents). In the small proportion of cases where multiple activities have been reported for a particular compound-target pair, a consensus value was selected as the median of those activities falling in the modal log unit. The simplified molecular-input line-entry system (SMILES) representation of the molecules was used to calculate molecular properties such as molecular weight (MW), logarithm of the partition coefficient (LogP), topological polar surface area (TPSA), etc. For this we used Dragon version 6 (Mauri et al. 2006), which is a commercially available software library that can potentially calculate up to 4885 molecular descriptors, depending on the availability of 3D structural information of the molecules. A full list is available on Dragon's website.[4]

As ChEMBL records 2D molecular structures only, we were restricted to estimating a maximum of 1447 molecular descriptors. We decided to generate datasets using all permitted molecular descriptors as features, and then to extract a subset of 43, which Dragon identifies as basic or constitutional descriptors. We call these representations 'allmolprop' and 'basicmolprop', respectively. For some of the molecules, Dragon failed to compute some of the descriptors, possibly because of bad or malformed structures, and these were treated as missing values. To avoid favouring QSAR algorithms able to deal with missing values, we decided to impute them, as a preprocessing step, using the median value of the corresponding feature.

In addition, we calculated the FCFP4 fingerprint representation using the Pipeline Pilot software from BIOVIA (Rogers and Hahn 2010). The fingerprint representation is the most commonly used in QSAR learning, whereby the presence or absence of a particular molecular substructure in a molecule (e.g. methyl group, benzine ring) is indicated by a Boolean variable. The FCFP4 fingerprint implementation generates 1024 such Boolean variables. We call this dataset representation 'fpFCFP4'. All of the fpFCFP4 datasets were complete, so a missing value imputation step is not necessary.

In summary, we use 3 types of feature representations and 1 level of preprocessing, thus generating 3 different dataset representations for each of the QSAR problems (targets), see Table 2. This produced in total 8292 datasets from the 2764 targets.

2.3 Baseline QSAR experiments

The predictive performance of all the QSAR learning methods on the datasets (base QSAR experiments) was assessed by taking the average root mean squared error (RMSE) with tenfold cross-validation.

[4] http://www.talete.mi.it.

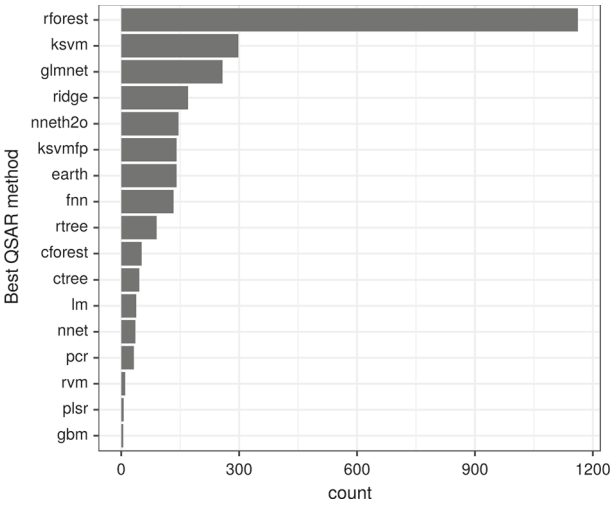
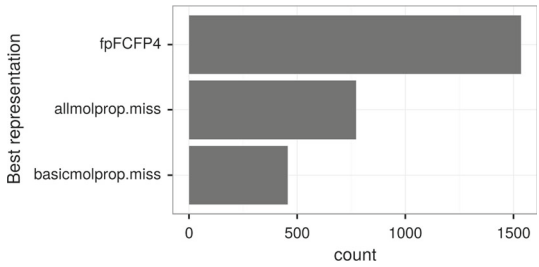


Fig. 2 Graphical representation of the number of times (target counts) a particular QSAR learning method obtains the best performance (minimum RMSE)

Fig. 3 Graphical representation of the number of times (target counts) a dataset representation was fitted with the best performer QSAR method (minimum RMSE)



We used the parameter settings mentioned in Table 1 for all experiments. Figure 2 summarizes the overall relative performance (in frequencies) of the QSAR methods for all dataset representations previously mentioned in Table 2. Results showed that random forest ('rforest') was the best performer in 1162 targets out of 2764, followed by SVM ('ksvm'), 298 targets, and GLM-NET ('glmnet'), 258 targets. In these results, the best performer is the algorithm with the lowest RMSE, even if it wins by a small margin. In terms of dataset representation, it turned out that datasets formed using FCFP4 fingerprints yielded consistently better models than the rest of the datasets (in 1535 out of 2764 situations). Results are displayed in Fig. 3.

Figure 4 summarizes the results obtained using various strategies (combinations of QSAR algorithm and dataset representation). As the figure shows, the bar plot is highly skewed towards the top ranked QSAR strategies with a long tail representing QSAR problems in which other algorithms perform better.

Applying random forest to datasets formed using either FCFP4 fingerprints or all molecular properties were the most successful QSAR strategies (in the figure, rforest.fpFCFP4 for 675 and rforest.allmolprop.miss for 396 out of 2764 targets, respectively). Other strategies, such as regression with ridge penalisation (ridge.fpFCFP4), SVM with Tanimoto kernel (ksvmfp.fpFCFP4), and SVM with RBF kernel (ksvm.fpFCFP4) were particularly successful when using the FCFP4 fingerprint dataset representation (for 154, 141, and 126 targets,

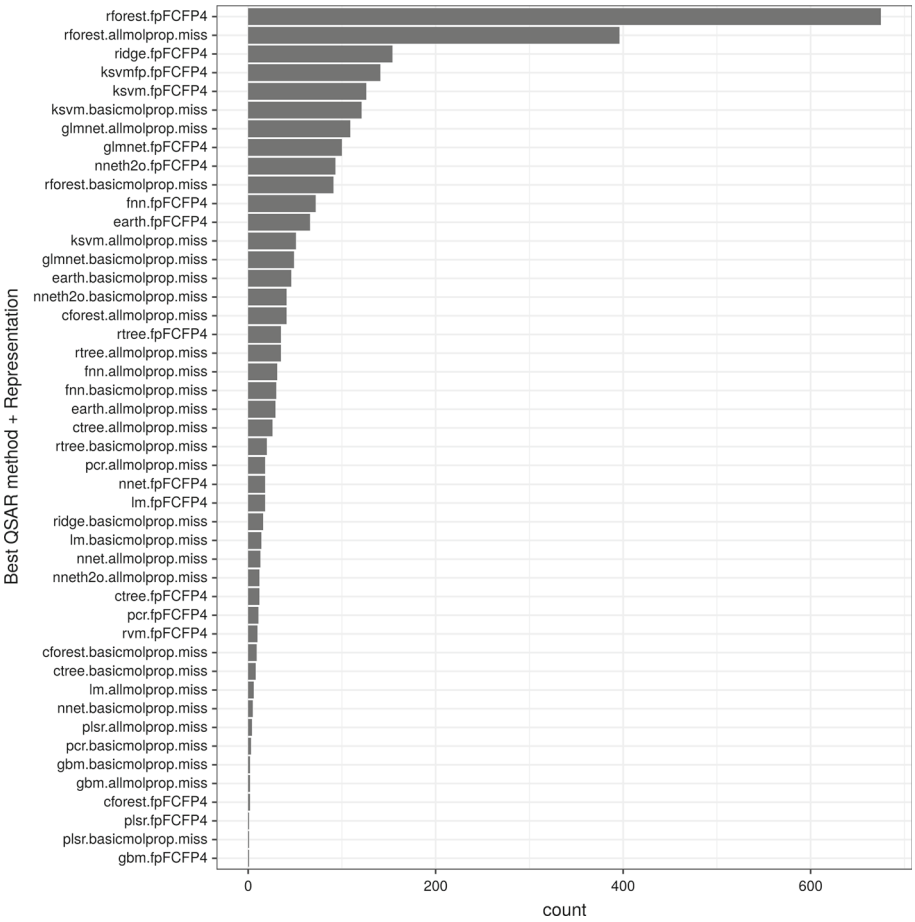


Fig. 4 Graphical representation of the number of times (target counts) a combination of dataset representation and QSAR method obtained the best performance (minimum RMSE)

respectively). The full list of strategies ranked by frequency of success is shown in the figure. Combinations that never produced best performances are not shown.

Combinations of QSARs and representations were also ranked by their average performances. For this, we estimated an average RMSE ratio score (aRMSEr) which is adapted from Brazdil et al. (2003), originally introduced for classification tasks. Our score was formulated as follows:

$$aRMSEr_p = \frac{\sum_q \sqrt[n]{\prod_i RMSEr^i_{p,q}}}{m}$$

where $RMSEr^i_{p,q} = RMSE^i_q / RMSE^i_p$ is the (inverse) RMSE ratio between algorithms p and q for the dataset i. In the same equation, m represents the number of algorithms, whilst n, the number of targets. Notice that, an $RMSEr^i_{p,q} > 1$ indicates that algorithm p outperformed algorithm q. Ranking results using aRMSEr are presented in Fig. 5.

We ran a Friedman test with a corresponding pairwise post-hoc test (Demsar 2006), which is a non-parametric equivalent of ANOVA in order to verify whether the performances of

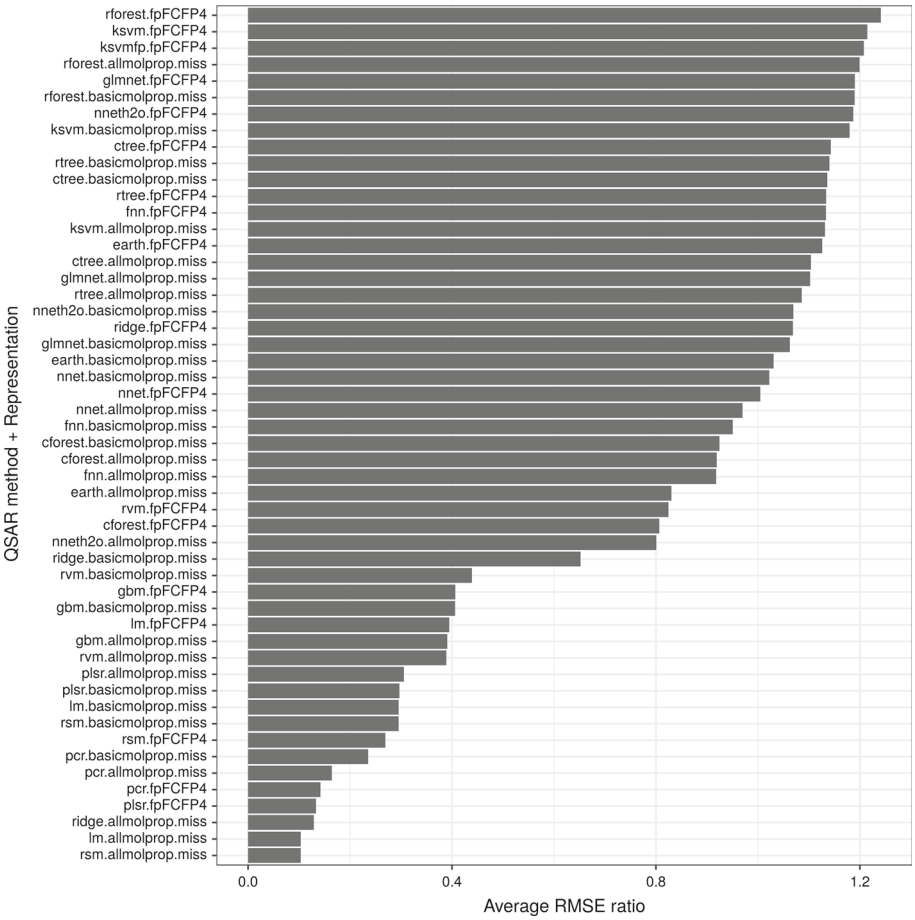


Fig. 5 Average ranking of dataset representation and QSAR combination as estimated using the RMSE ratio

baseline QSAR strategies were statistically different. The Friedman test ranks the strategies used per dataset according to their performance and tests them against the null hypothesis that they are equivalent. A post-hoc test was carried out if the null hypothesis is rejected. For this we used the Nemenyi test, also suggested by Demsar (2006). The resulting P value ($10E-06$) from the test indicates the null hypothesis was invalid (P value $<<$ 0.05), which suggests that algorithm selection should significantly impact the overall performance.

We ran the aforementioned post-hoc test for the top 6 QSAR strategies[5] presented in Fig. 5. Results are shown in Fig. 6. It shows that performance differences between the QSAR strategies were statistically significant with the exception of rforest.allmolprop.miss versus ksvmfp.fpFCFP4.

[5] Testing all possible pairwise combinations of QSAR strategies was not possible as the post-hoc test was running extremely slowly and we considered it would not add to the analyses of the results.

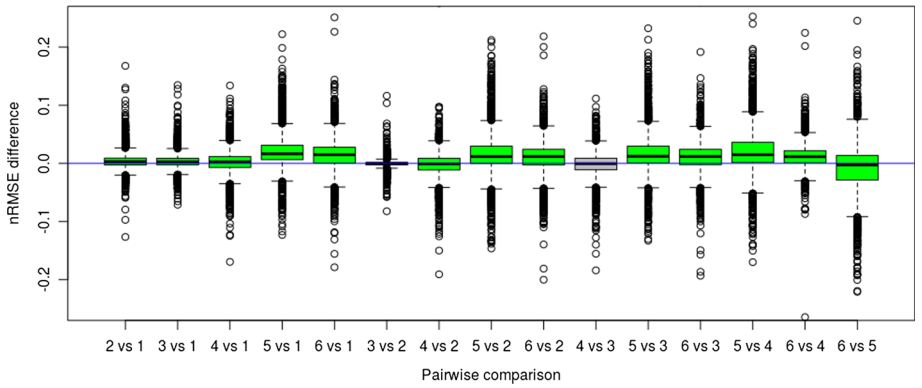


Fig. 6 Box plot displays the post-hoc test results over the top 6 ranked best performer QSAR strategies: 1—rforest.fpFCFP4, 2—ksvm.fpFCFP4, 3—ksvmfp.fpFCFP4, 4—rforest.allmolprop.miss, 5—glmnet.fpFCFP4, and 6—rforest.basicmolprop.miss.fs. Statistically significant comparisons (P value < 0.05) represented with green boxes (Color figure online)

3 Meta-features for meta-QSAR learning

3.1 Meta-QSAR ontology

Meta-learning analysis requires a set of meta-features. In our meta-QSAR study we used as meta-features, characteristics of the datasets considered in the base study and drug target properties. We utilised a similar approach employed by BODO (the Blue Obelisk Descriptor Ontology) (Floris et al. 2011) and the Chemical Information Ontology (Hastings et al. 2011) for the formal definitions of molecular descriptors used in QSAR studies, and developed a meta-QSAR ontology.[6]

The meta-QSAR ontology provides formal definitions for the meta-features used in the reported meta-QSAR study (see Fig. 7). The meta-features are defined at the conceptual level, meaning that the ontology does not contain instance-level values of meta-features for each of 8292 considered dataset. For example, the meta-feature 'multiple information' is defined as the meta-feature of a dataset (multiple information (also called total correlation) among the random variables in the dataset), but the meta-QSAR ontology does not contain values of this meta-feature for each dataset. Instead, it contains links to the code to calculate values of the relevant features. For example, we used the R Package Peptides[7] to calculate values of the meta-feature 'hydrophobicity'. Figure 8 shows how this information is captured in the meta-QSAR ontology. The description of the selected meta-features and instructions on the calculation of their values are available online.[8]

3.2 Dataset meta-features

The considered 8292 datasets have a range of different properties, e.g. 'number of compounds' (instances) in the dataset, 'entropy' and 'skewness' of the features and 'target meta-feature', 'mutual information' and 'total correlation' between the input and output features (see Table 3

[6] The ontology is available at https://github.com/larisa-soldatova/meta-qsar.

[7] http://cran.r-project.org/web/packages/Peptides/.

[8] https://github.com/meta-QSAR/drug-target-descriptors.

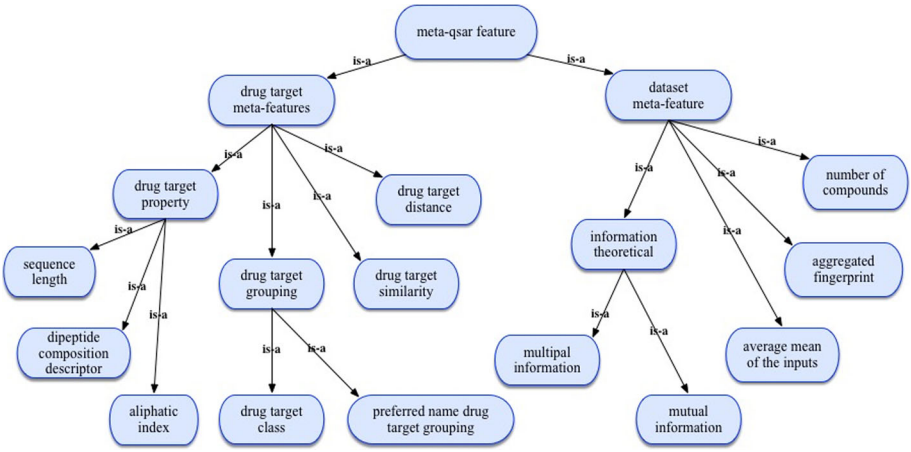


Fig. 7 The key branches of the meta-QSAR ontology (a fragment)

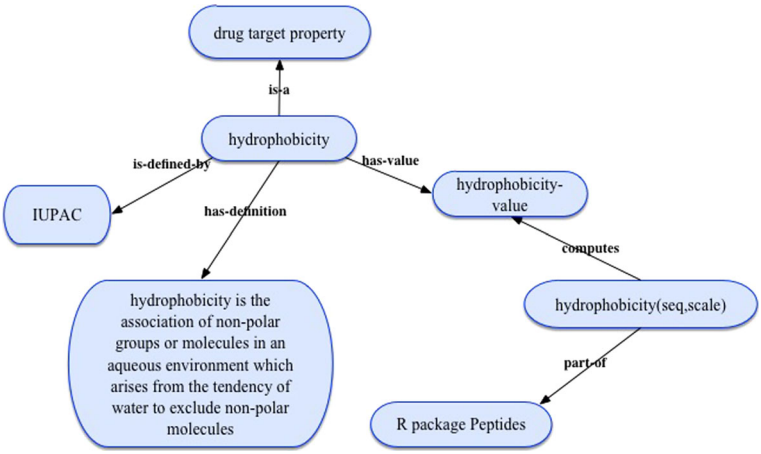


Fig. 8 The representation of the meta-features and their values

for more detail). The dataset properties have a significant effect on the performance of the explored algorithms and were used for the meta-QSAR learning. Figure 11 shows the level of influence of different categories of meta-features. For example information-theoretical meta-features make a considerable contribution to meta-learning.

Some descriptors of the dataset properties, e.g. 'number of instances', have been imported from the Data Mining Optimization (DMOP) Ontology[9] (Keeta et al. 2015). We also added QSAR-specific dataset descriptors 'aggregated fingerprint'. These were calculated by summing 1s (set bits) in each of the 1024 columns and normalised by the number of the compounds in each dataset.

[9] www.dmo-foundry.org/DMOP.

Table 3 Dataset meta-features (examples)

Feature	Description
multiinfo	Multiple information (also called total correlation) among the random variables in the dataset
mutualinfo	Mutual information between nominal attributes X and Y. Describes the reduction in uncertainty of Y due to the knowledge of X, and leans on the conditional entropy $H(Y\|X)$
nentropyfeat	Normalised entropy of the features which is the class entropy divided by log(n) where n is the number of the features
mmeanfeat	Average mean of the features
msdfeat	Average standard deviation of the features
kurtresp	Kurtosis of the response variable
meanresp	Mean of the response variable
skewresp	Skewness of the response variable
nentropyresp	Normalised entropy of the response variable
sdresp	Standard deviation of the response
aggFCFP4fp (1024 features)	Aggregated fingerprints and normalized over the number of instances in the dataset

3.3 Drug target meta-features

3.3.1 Drug target properties

The QSAR datasets are additionally characterized by measurable properties of the drug target (a protein) they represent, such as 'aliphatic index', 'sequence length', 'isoelectric point' (see Table 4 for more details). These differ from the molecular properties we used to describe the chemical compounds in the QSAR dataset instances, e.g. 'molecular weight' (MW), 'LogP'.

3.3.2 Drug target groupings

We also used drug target groupings (Imming et al. 2006), such as 'drug target classes', and 'the preferred name groupings', as meta-features. These enable meta-learning to exploit known biological/chemical relationships between the targets (proteins). Indeed, if the target proteins are similar, this may make the resulting datasets more similar too.

Drug target classes The ChEMBL database curators have classified the protein targets in a manually curated family hierarchy. The version of the hierarchy that we have used (taken from ChEMBL20) comprises 6 levels, with Level 1 (L1) being the broadest class, and Level 6 (L6) the most specific. For example, the protein target 'Tyrosine-protein kinase Srms' is classified as follows: Enzyme (L1), Kinase (L2), Protein Kinase (L3), TK protein kinase group (L4), Tyrosine protein kinase Src family (L5), Tyrosine protein kinase Srm (L6). Different classes in Level 1 are not evolutionarily related to one another, whereas members of classes in L3 and below generally share common evolutionary origins. The picture is mixed for L2. The hierarchy is not fully populated, with the greatest emphasis being placed on the target families of highest pharmaceutical interest, and the different levels of the hierarchy are not

Table 4 Drug targets meta-features (examples)

Feature	Description
Aliphatic index	The Aliphatic index (Atsushi 1980) is defined as the relative volume occupied by aliphatic side chains (Alanine, Valine, Isoleucine, and Leucine). It may be regarded as a positive factor for the increase of thermo stability of globular proteins
Boman index	This the potential protein interaction index proposed by Boman (2003). It is calculated as the sum of the solubility values for all residues in a sequence (Rondn-Villarreal et al. 2014)
Hydrophobicity (38 features)	Hydrophobicity is the association of non-polar groups or molecules in an aqueous environment which arises from the tendency of water to exclude non-polar molecules (Mcnaught and Wilkinson 1997). We estimated 38 variants of hydrophobicity
Net charge	The theoretical net charge of a protein sequence as described by Moore (1985)
Molecular weight	Ratio of the mass of a molecule to the unified atomic mass unit. Sometimes called the molecular weight or relative molar mass (Mcnaught and Wilkinson 1997)
Isoelectric point	The pH value at which the net electric charge of an elementary entity is zero. (pI is a commonly used symbol for this kind of quantity, however, a more accurate symbol is pH(I)) (Mcnaught and Wilkinson 1997)
Sequence length	The number of amino acids in a protein sequence
Instability index	The instability index was proposed by (Guruprasad, 1990). A protein whose instability index is smaller than 40 is predicted as stable, a value above 40 predicts that the protein may be unstable
DC groups (400 features)	The Dipeptide Composition descriptor (Xiao et al. 2015; Bhasin and Raghava 2004) captures information about the fraction and local order of amino acids

defined by rigorous criteria. However, the hierarchical classification provides a useful means of grouping related targets at different levels of granularity.

The preferred name drug targets grouping The ChEMBL curators have also assigned each protein target a *preferred name*—in a robust and consistent manner, independent of the various adopted names and synonyms used elsewhere. This preferred name is based on the practice that individual proteins can be described by a range of different identifiers and textual descriptions across the various data resources. The detailed manual annotation of canonical target names means that, for the most part, orthologous proteins (evolutionarily related proteins with the same function) from related species are described consistently, allowing the most related proteins to be grouped together. In the preferred name groupings, we obtained 468 drug target groups, each with two or more drug targets. The largest drug target group is that of Dihydrofolate Reductase with 21 drug targets.

3.3.3 Drug target distances and similarities

Drug target distances We used the 6-level ChEMBL hierarchy tree to compute distances between target families as meta-features for the meta-QSAR learning. Assuming that each

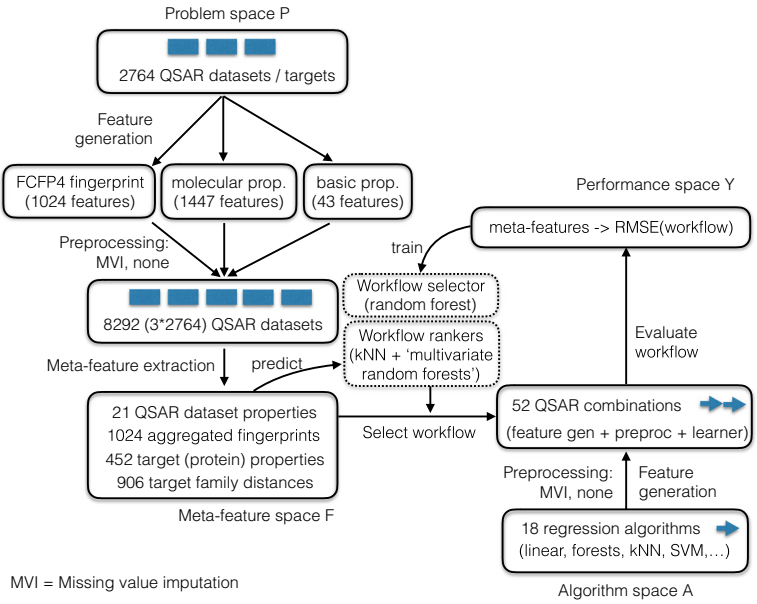


Fig. 9 Meta-learning setup to select QSAR combinations (workflows) for a given QSAR dataset. The 52 QSAR combinations are generated by combining 3 types of representation/preprocessing with 17 regression algorithms, plus the Tanimoto KSVM which was only run on the fingerprint representation

target family corresponds to a node in the tree, the distance between two family targets is defined as the shortest path between their respective two nodes. We developed an R package to build the tree and compute the distance between the nodes.[10]

4 Meta-learning: QSAR algorithm selection

We cast the meta-QSAR problem as two different problems: (1) the classification task to predict which QSAR method should be used for a particular QSAR problem; and (2) ranking prediction task to rank QSAR methods by their performances. This entails a number of extensions to Rice's framework in Fig. 1, as we are now dealing with multiple dataset representations per QSAR problem, and learning algorithm. The resulting setup is shown in Fig. 9. Each original QSAR problem is first represented in 3 different ways resulting in 3 datasets for each QSAR target, from which we extract 11 dataset-based meta-features each (see Sect. 3.2),[11] as well as over 450 meta-features based on the target (protein) that the dataset represents (see Sect. 3.3). The space of algorithms consists of workflows that generate the base-level features, and run one of the the 18 regression algorithms (see Sect. 2.1), resulting in 52 workflows which are evaluated, based on their RMSE, on the corresponding datasets (those with the same representation).

[10] The R package is available at: https://github.com/meta-QSAR/simple-tree.

[11] The actual number (21) is slightly smaller because some meta-features, such as the number of instances, is identical for each dataset.

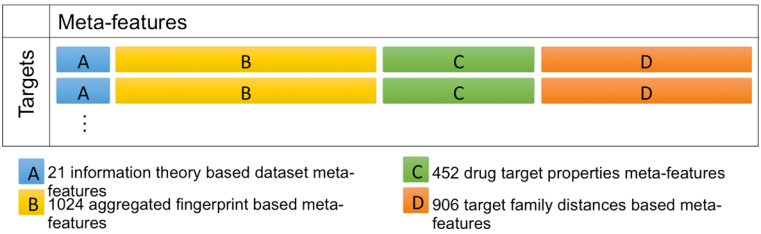


Fig. 10 Schematic representation of the meta-dataset used for meta-QSAR

4.1 Meta-QSAR dataset

A training meta-dataset was formed using the meta-features extracted from the baseline QSAR datasets as the inputs. For the classification tasks we used the best QSAR strategy (combination of QSAR method and dataset representation) per target as the output labels, whilst for the ranking tasks, the QSAR performances (RMSEs) were used. Figure 10 shows a schematic representation of the meta-dataset used in the meta-learning experiments. As this figure shows, we used meta-features derived from dataset and drug target properties. The size of the final meta-dataset was 2394 meta-features by 2764 targets.

4.2 Meta-QSAR learning algorithms

A meta-learning classification problem using all possible combinations of QSAR methods and dataset representions was implemented using a random forest with 500 trees. Given the large number of classes (52 combinations) and the highly imbalanced classification problem (as shown in Fig. 4, additional random forest implementations using the top 2, 3, 6, 11 and 16 combinations (Fig. 5) were also investigated. For the ranking problem, we used two approaches: K-nearest neighbour approach (k-NN), as suggested in Brazdil et al. (2003), and a multi-target regression approach. Experiments with k-NN were carried out using 1, 5, 10, 50, 100, 500, and all neighbours. The multi-target regression was implemented using a multivariate random forest regression (Segal and Xiao 2001) with 500 trees to predict QSAR performances and with them, to rank QSAR combinations. All implementations were assessed using tenfold cross-validation.

4.3 Results

We used the all-classes random forest implementation to estimate the importance of each meta-feature in the classification task, as estimated using the mean decrease accuracy. Summary results considered by meta-feature groups are presented in Fig. 11. It is seen that the meta-features belonging to the information theory group (all dataset meta-features but the aggregated fingerprints, Table 3) were the most relevant, although we found all groups contributed to the task.

As mentioned before, k-NN and multivariate random forest were used to implement ranking models. We used the Spearman's rank correlation coefficient to compare the predicted with the actual rankings (average of the actual rankings were shown in Fig. 5). Results of these comparisons are shown in Fig. 12. It is observed from the figure that the multivariate random forest and 50-nearest neighbours implementations (mRF and 50-NN in the figure) predicted

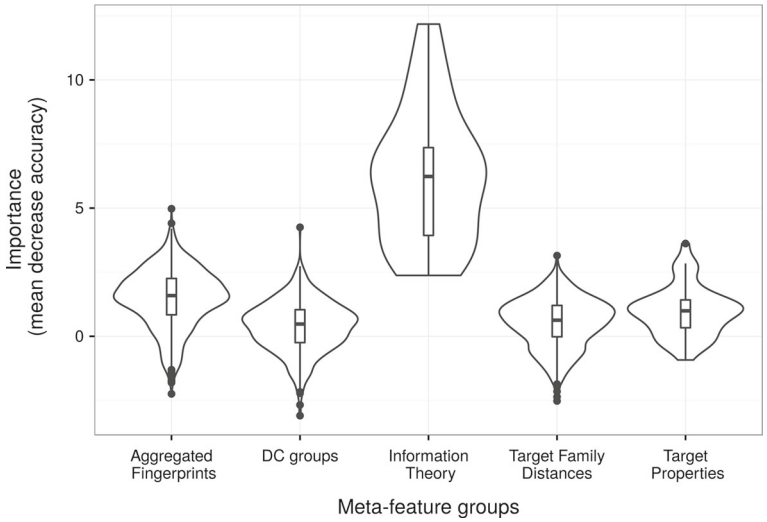


Fig. 11 Violin plots with added box plots representing the mean decrease accuracy of the meta-features grouped by meta-feature groups. Notice that for visualization purposes, we are showing the group dataset meta-features (as defined in Sect. 3) in two separated groups: "Aggregated Fingerprints" and "Information Theory"

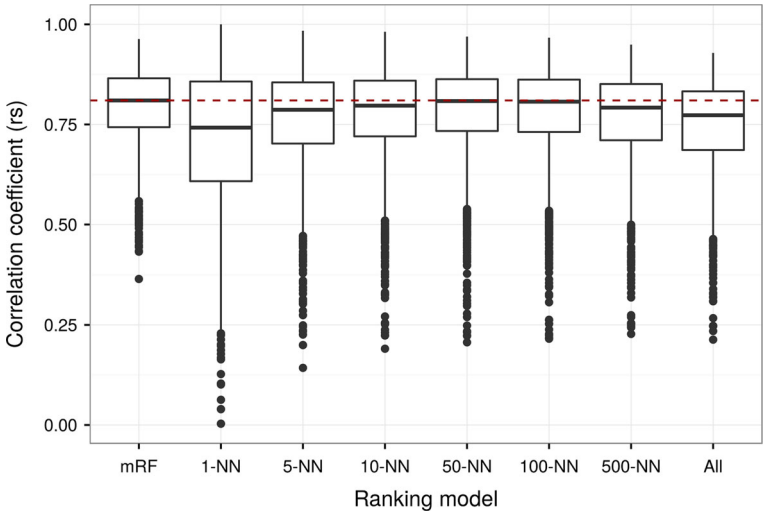


Fig. 12 Box plots representing the computed Spearman's rank correlation coefficient (rs) between the predicted and actual rankings. Labels in the horizontal axis indicates: mRF—multivariate random forest, 1-NN, 5-NN, 10-NN, 50-NN, 100-NN, 500-NN, and All—1, 5, 10, 50, 100, 500, and all nearest neighbours, respectively

better rankings, overall. For illustrative purpose, the average of the predicted rankings by multivariate random forest is displayed in Fig. 13.

Performances of the best suggested QSAR combination by all Meta-QSAR implementations were compared with an assumed default. In the case of the ranking models, the best suggested QSAR combination is the one ranked the highest in each QSAR problem. For

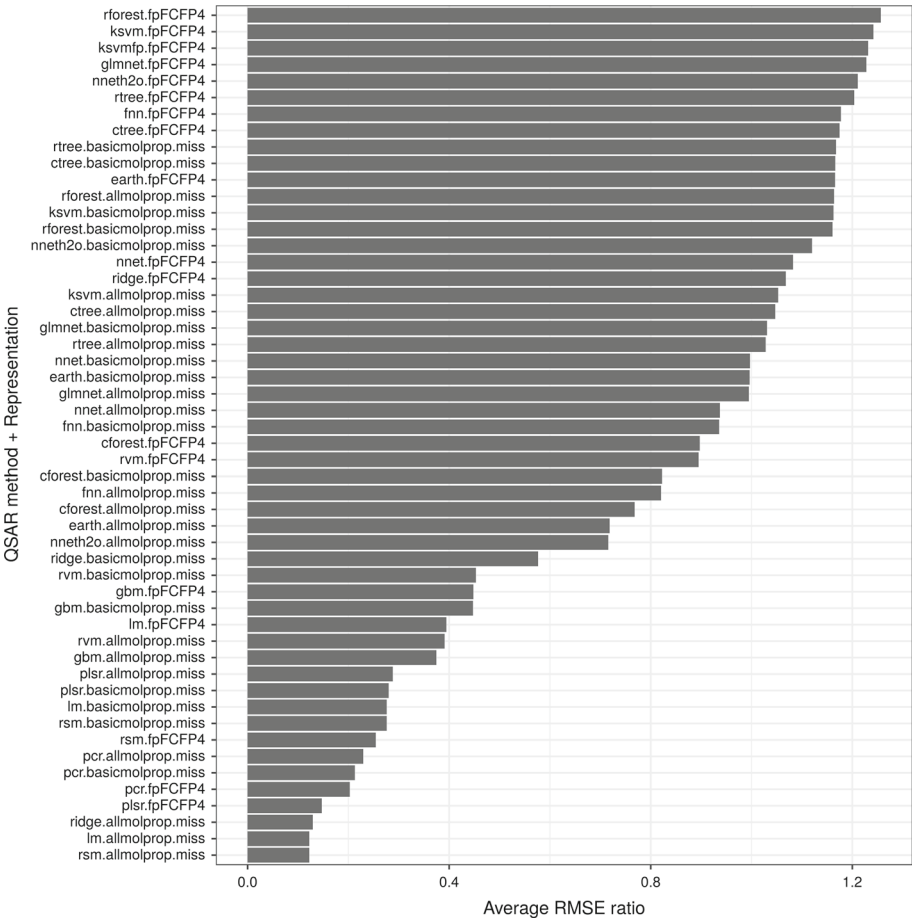


Fig. 13 Average of predicted ranking of QSAR combinations using the multivariate random forest algorithm according to the RMSE ratio

the default (baseline) we used random forest with the fingerprint molecular representation (rforest.fpFCFP4), as this is well-known for its robust and reliable performance (Fig. 4), and hence represents a strong baseline. Results are shown in Fig. 14. As can be observed in this figure, most of the Meta-QSAR implementations improved overall performance in comparison with the default QSAR combination with the exception of the 1-nearest neighbour. The same results are summarized in Table 5. We ran a Wilcoxon Rank Sum tests on the RMSE differences between the Meta-QSAR implementations and the assumed default. Results, in the form of *P* values, are also shown in the same table. According to these tests, performance improvements by Meta-QSAR implementations *cl.Top11*, *cl.Top16*, and *cl.All* were not statistically significant (P value > 0.05). Overall, the results suggest that meta-learning can be successfully used to select QSAR algorithm/representation pairs that perform better than the best algorithm/representation pair (default strategy).

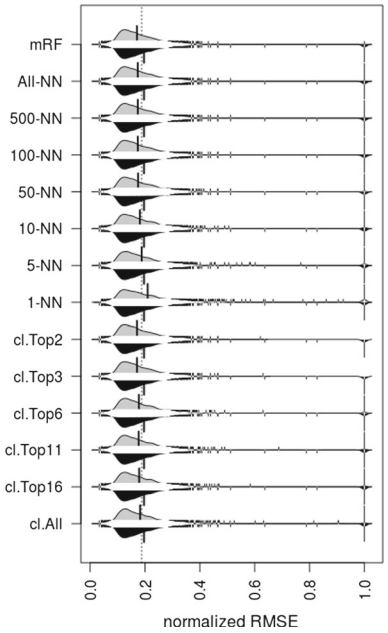


Fig. 14 Visual comparison of performance distributions between the default strategy (in black) and all meta-learners (in grey) using asymmetric bean plots. Average RMSE for each implementation is represented by vertical black lines on the "beans" (performance distribution curves)

Table 5 Comparison of performance between the default strategy and all Meta-QSAR implementations

Implementation	Mean RMSE	Relative RMSE reduction (%)	*P* value
Default	0.1964		
mRF	0.1709	13.0	< 0.001
All-NN	0.1737	11.6	< 0.001
500-NN	0.1738	11.5	< 0.001
100-NN	0.1738	11.5	0.009
50-NN	0.1751	10.9	0.003
10-NN	0.1815	7.6	0.011
5-NN	0.1881	4.3	< 0.001
1-NN	0.2098	-6.8	< 0.001
cl.Top2	0.1711	12.9	0.007
cl.Top3	0.1709	13.0	< 0.001
cl.Top6	0.1779	9.5	0.022
cl.Top11	0.1771	9.9	0.086
cl.Top16	0.1788	9.0	0.072
cl.All	0.1823	7.2	0.189

Relative RMSE reduction (in%) is estimated as $(mRMSE_{Def} - mRMSE_{MQSAR})/mRMSE_{Def} \times 100\%$, where $mRMSE_{Def}$ and $mRMSE_{MQSAR}$ correspond to the mean RMSE of the default and Meta-QSAR strategies, respectively. *P* values were estimated using Wilcoxon Rank Sum test

5 Discussion

QSAR models are regression models, empirical functions that relate a quantitative description of a chemical structure (a drug) to some form of biological activity (e.g. inhibiting proteins) for the purposes of informing drug design decision-making. Many consider the seminal papers of Hansch and Fujita (1964) to be the origin of the QSAR field. Since then, such

predictive modelling approaches have grown to become a core part of the drug discovery process (Cumming et al. 2013; Cherkasov et al. 2014). The subject is still increasing in importance (Cramer 2012). This may be attributed to the alignment of a number of factors, including increased availability of data, advances in data-mining methodologies as well as a more widespread appreciation of how to avoid many of the numerous pitfalls in building and applying QSAR models (Cherkasov et al. 2014). Current trends in the field include efforts in chemical data curation (Williams et al. 2012), automation of QSAR model building (Cox et al. 2013), exploration of alternative descriptors (Cherkasov et al. 2014), and efforts to help define the Applicability Domain (AD) of a given QSAR model (Sahigara et al. 2012).

To facilitate application of QSAR models in the drug regulatory process, the Organization for Economic Co-operation and Development (OECD) has provided guidance to encourage good practice in QSAR modelling. The OECD guidelines recommend that a QSAR model has (i) a defined end point; (ii) an unambiguous algorithm; (iii) a defined domain of applicability; (iv) appropriate measures of goodness of fit, robustness and predictivity; and (v) a mechanistic interpretation, if possible. However, the application of QSAR models in drug discovery is still fraught with difficulties, not least because the model builder is faced with myriad options with respect to choice of descriptors and machine learning methods.

The application of meta-learning in this study helps ameliorate this issue by providing some guidance as to which individual method performs the best overall as well as which method may be the most appropriate given the particular circumstances.

Our comparison of QSAR learning methods involves 18 regression methods and 3 molecular representations applied to more than 2700 QSAR problems, making it one of the most extensive ever comparisons of base learning methods reported. Moreover, the QSAR datasets, source code, and all our experiments are available on OpenML (Vanschoren et al. 2013),[12] so that our results can be easily reproduced. This is not only a valuable resource for further work in drug discovery, it will foster the development of meta-learning methods as well. Indeed, as all the experimental details are fully available, there is no need to run the baseline-learners again, so research effort can be focused on developing novel meta-learning methods.

In this paper we have investigated algorithm selection for QSAR learning. Note however, that many more meta-learning approaches could be applied: it would be interesting to investigate other algorithm selection methods (see Sect. 1.2.2), such as other algorithm ranking approaches (e.g. active testing or collaborative filtering), and model-based optimization. Another alternative framing of the meta-learning problem would be to use a regression algorithm at the meta-level and predict the *performance* of various regression algorithms. We will explore this in future work. Finally, we would also like to explore other algorithm selection techniques beyond Random Forests. To this end, we plan to export our experiments from OpenML to an ASlib scenario (Bischl et al. 2016), where many algorithm selection techniques could be compared.

The success of meta-learning crucially depends on having a large set of datasets to train a meta-learning algorithm, or simply to find similar prior datasets from which best solutions could be retrieved. This work provides more than 8000 datasets, which is several orders of magnitude larger than what was available before. It has often been observed that machine learning breakthroughs are being made by having novel large collections of data: ImageNet,[13] for instance, sparked breakthroughs in image recognition with deep learning. The datasets made available here could have a similar effect in accelerating meta-learning research, as well as novel machine learning solutions for drug discovery. Moreover, it is but the first example

[12] See http://www.openml.org/s/13.

[13] http://www.image-net.org.

of what is possible if large collections of scientific data are made available as readily usable datasets for machine learning research. Beyond ChEMBL, there exist many more databases in the life sciences and other fields (e.g. physics and astronomy), which face similar challenges in selecting the best learning algorithms, hence opening up interesting further avenues for meta-learning research.

Beyond the number of datasets, this study pushes meta-learning research in several other ways. First, it is one of the few recent studies focussing on regression problems rather than classification problems. Second, it uses several thousands (often domain-specific) meta-features, which is much larger than most other reported studies. And third, it considers not only single learning algorithms, but also (small) workflows consisting of both preprocessing and learning algorithms.

There is ample opportunity for future work. For instance, besides recommending the best algorithm, one could recommend the best hyperparameter setting as well (e.g. using model-based optimization). Moreover, we did not yet include several types of meta-features, such as landmarkers or model-based meta-features, which could further improve performance. Finally, instead of using a RandomForest meta-learner, other algorithms could be tried as well. One particularly interesting approach would be to use Stacking (Wolpert 1992) to combine all the individually learned models into a larger model that exploits the varying quantitative predictions of the different base-learner and molecular representation combinations. However developing such a system is more computationally complex than simple algorithm selection, as it requires applying cross-validation over the base learners.

6 Conclusions

QSAR learning is one of the most important and established applications of machine learning. We demonstrate that meta-learning can be leveraged to build QSAR models which, on average, could improve performance by up to 13% with regard to those learned with any base-level regression algorithm. We carried out the most comprehensive ever comparison of machine learning methods for QSAR learning: 18 regression methods, 3 molecular representations, applied to more than 2700 QSAR problems. This enabled us to first compare the success of different base-learning methods, and then to compare these results with meta-learning. We found that algorithm selection significantly outperforms the best individual QSAR learning method (random forests using a molecular fingerprint representation). The application of meta-learning in this study helps accelerate research in drug discovery by providing guidance as to which machine learning method may be the most appropriate given particular circumstances. Moreover, it represents one of the most extensive meta-learning studies ever, including over 8000 datasets and several thousands of meta-features. The success of meta-learning in QSAR learning provides evidence for the general effectiveness of meta-learning over base-learning, and opens up novel avenues for large-scale meta-learning research.

Acknowledgements This research was funded by the UK Engineering and Physical Sciences Research Council (EPSRC) Grant EP/K030469/1.

References

Abdulrahman, S., & Brazdil, P. (2014). Measures for combining accuracy and time for meta-learning. In *Proceedings of the international workshop on meta-learning and algorithm selection co-located with 21st European conference on artificial intelligence, MetaSel@ECAI 2014, Prague, Czech Republic, August 19, 2014* (pp. 49–50).

Amasyali, M. F., & Ersoy, O. K. (2009). A study of meta learning for regression. Research report, Purdue University. http://docs.lib.purdue.edu/ecetr/386.

Atsushi, I. (1980). Thermostability and aliphatic index of globular proteins. *Journal of Biochemistry*, *88*(6), 1895–1898.

Bardenet, R., Brendel, M., Kégl, B., & Sebag, M. (2013). Collaborative hyperparameter tuning. In S. Dasgupta & D. McAllester (Eds.), *30th international conference on machine learning (ICML 2013)* (Vol. 28 , pp. 199–207). Acm Press. http://hal.in2p3.fr/in2p3-00907381.

Bensusan, H., & Giraud-Carrier, C. (2000). Casa batló is in passeig de gràcia or landmarking the expertise space. *Proceedings of the ECML-00 workshop on meta-learning: Building automatic advice strategies for model selection and method combination* (pp. 29–46).

Bensusan, H., & Kalousis, A. (2001). Estimating the predictive accuracy of a classifier. *Lecture Notes in Computer Science*, *2167*, 25–36.

Bhasin, M., & Raghava, G. P. S. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. *Journal of Biological Chemistry*, *279*(22), 23262–23266.

Bickel, S., Bogojeska, J., Lengauer, T., & Scheffer, T. (2008). Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on machine learning*, ICML '08, pp. 56–63, New York, NY, USA. ACM. ISBN: 978-1-60558-205-4. https://doi.org/10.1145/1390156.1390164.

Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Frechtte, A., et al. (2016). Aslib: A benchmark library for algorithm selection. *Artificial Intelligence Journal*, *237*, 41–58.

Bischl, B., Mersmann, O., Trautmann, H., & Preuss, M. (2012). Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proceedings of the fourteenth annual conference on genetic and evolutionary computation* (pp. 313320).

Boman, H. G. (2003). Antibacterial peptides: Basic facts and emerging concepts. *Journal of internal medicine*, *254*(3), 197–215.

Braun, L. A., Tiralongo, E., Wilkinson, J. M., Poole, S., Spitzer, O., Bailey, M., et al. (2010). Adverse reactions to complementary medicines: The Australian pharmacy experience. *International Journal of Pharmacy Practice*, *18*(4), 242–244.

Brazdil, P., & Soares, C. (2000). *Ranking classification algorithms based on relevant performance information*. In *Meta-learning: Building automatic advice strategies for model selection and method combination*.

Brazdil, P., Soares, C., & Da Costa, J. P. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, *50*, 251–277.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599.

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., et al. (2014). QSAR modeling: Where have you been? Where are you going to? *Journal of Medicinal Chemistry*, *57*(12), 4977–5010.

Cox, R., Green, D. V. S., Luscombe, C. N., Malcolm, N., & Pickett, S. D. (2013). QSAR workbench: Automating QSAR modeling to drive compound design. *Journal of Computer-Aided Molecular Design*, *27*(4), 321–336.

Cramer, R. D. (2012). The inevitable QSAR renaissance. *Journal of Computer-Aided Molecular Design*, *26*(1), 35–38.

Cumming, J. G., Davis, A. M., Muresan, S., Haeberlein, M., & Chen, H. (2013). Chemical predictive modelling to improve compound quality. *Nature Reviews Drug Discovery*, *12*(12), 948–962.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

DiMasi, J. A., Grabowski, H. G., & Hansen, R. W. (2015). The cost of drug development [letter to the editor]. *New England Journal of Medicine*, *372*(20), 1972.

dos Santos, P., Ludermir, T., & Prudêncio, R. (2004). Selection of time series forecasting models based on performance information. *Proceedings of the 4th international conference on hybrid intelligent systems* (pp. 366–371).

Feurer, M., Springenberg, T., & Hutter, F. (January 2015). Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the twenty-ninth AAAI conference on artificial intelligence*.

Floris, M., Willighagen, E., Guha, R., Rojas, M., & Hoppe, C. (2011). The Blue Obelisk descriptor ontology. Available at: http://qsar.sourceforge.net/dicts/qsar-descriptors/index.xhtml.

Fürnkranz, J., & Petrak, J. (2001). An evaluation of landmarking variants. *Working notes of the ECML/PKDD 2001 workshop on integrating aspects of data mining, decision support and meta-learning* (pp. 57–68).

Guerri, A., & Milano, M. (2012). Learning techniques for automatic algorithm portfolio selection. In *Proceedings of the sixteenth european conference on artificial intelligence* (pp. 475479).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009) The weka data mining software: An update. *SIGKDD Explor. Newsl., 11*(1), 10–18. ISSN: 1931-0145. https://doi.org/10.1145/1656274.1656278.

Hansch, C., & Fujita, T. (1964). p-σ-π analysis. A method for the correlation of biological activity and chemical structure. *Journal of the American Chemical Society*, *86*(8), 1616–1626.

Hastings, J., Chepelev, L., Willighagen, E., Adams, N., Steinbeck, C., & Dumontier, M. (2011). The chemical information ontology: Provenance and disambiguation for chemical data on the biological semantic web. *Plos One*, *6*(10), e25513.

Hilario, M., & Kalousis, A. (2001). Fusion of meta-knowledge and meta-data for case-based model selection. *Lecture Notes in Computer Science*, *2168*, 180–191.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Proceedings of the conference on learning and intelligent optimization (LION 5)* (pp. 507–523).

Imming, P., Sinning, C., & Meyer, A. (2006). Drugs, their targets and the nature and number of drug targets. *Nature Reviews Drug Discovery, 5*(10), 821–834. ISSN: 1474-1776. https://doi.org/10.1038/nrd2132.

Ioset, J. R., & Chang, S. (2011). Drugs for Neglected Diseases initiative model of drug development for neglected diseases: Current status and future challenges. *Future Medicinal Chemistry, 3*(11), 1361–1371. https://doi.org/10.4155/fmc.11.102.

Kalousis, A. (2002). *Algorithm selection via meta-learning*. Ph.D. Thesis. University of Geneva.

Kalousis, A., & Hilario, M. (2001). Model selection via meta-learning: A comparative study. *International Journal on Artificial Intelligence Tools*, *10*(4), 525–554.

Keeta, C., Lawrynowiczb, A., d'Amatoc, C., et al. (2015). The data mining optimization ontology. *Journal of Web Semantics*, *32*, 43–53.

Köpf, C., Taylor, C., & Keller, J. (Jan 2000). Meta-analysis: From data characterisation for meta-learning to meta-regression. *Proceedings of the PKDD2000 workshop on data mining, decision support, meta-learning an ILP: Forum for practical problem representaion and prospective solutions* (pp. 15–26).

Lee, J. W., & Giraud-Carrier, C. G. (2008). Predicting algorithm accuracy with a small set of effective meta-features. In *Seventh international conference on machine learning and applications, ICMLA 2008, San Diego, CA, USA, 11–13 December 2008* (pp. 808–812).

Lee, J. W., & Giraud-Carrier, C. G. (2011). A metric for unsupervised metalearning. *Intelligent Data Analysis*, *15*(6), 827–841.

Leite, R.,& Brazdil, P. (2005). Predicting relative performance of classifiers from samples. *Proceedings of the 22nd international conference on machine learning* (pp. 497–504).

Leite, R., & Brazdil, P. (2007). An iterative process for building learning curves and predicting relative performance of classifiers. *Lecture Notes in Computer Science*, *4874*, 87–98.

Leite, R., Brazdil, P., & Vanschoren, J. (2012). Selecting classification algorithms with active testing. In *Machine learning and data mining in pattern recognition—8th international conference, MLDM 2012, Berlin, Germany, July 13–20, 2012. Proceedings* (pp. 117–131).

Ler, D., Koprinska, I., & Chawla, S. (2005). Utilizing regression-based landmarkers within a meta-learning framework for algorithm selection. *Technical report number 569 School of Information Technologies University of Sydney* (pp. 44–51).

Leslie, D. L., & Inouye, S. K. (2011). The importance of delirium: Economic and societal costs. *Journal of the American Geriatrics Society*, *59*(Suppl 2), S241–S243.

Lindner, G., & Studer, R. (1999). Ast: Support for algorithm selection with a cbr approach. In *Proceedings of the international conference on machine learning, workshop on recent advances in meta-learning and future work*.

Martin, Y. C. (2010). Tautomerism, Hammett sigma, and QSAR. *Journal of Computer-Aided Molecular Design*, *24*(6–7), 613–616.

Mauri, A., Consonni, V., Pavan, M., & Todeschini, R. (2006). Dragon software: An easy approach to molecular descriptor calculations. *MATCH Communications in Mathematical and in Computer Chemistry*, *56*, 237–248.

Mcnaught, A. D., & Wilkinson, A. (1997). *IUPAC. Compendium of chemical terminology, 2nd ed. (the "Gold Book").* New York: Wiley; 2nd Revised edition edition.

Misir, M., & Sebag, M. (2013). *Algorithm selection as a collaborative filtering problem*. Research report, INRIA. https://hal.inria.fr/hal-00922840.

Moore, D. S. (1985). Amino acid and peptide net charges: A simple calculational procedure. *Biochemical Education*, *13*(1), 10–11.

Pammolli, F., Magazzini, L., & Riccaboni, M. (2011). The productivity crisis in pharmaceutical R&D. *Nature Reviews Drug Discovery, 10*(6), 428–438.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, Y., Flach, P., Brazdil, P., & Soares, C. (2002). Decision tree-based data characterization for meta-learning. *ECML/PKDD'02 workshop on integration and collaboration aspects of data mining, decision support and meta-learning* (pp. 111–122).

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Tell me who can learn you and I can tell you who you are: landmarking various learning algorithms. In *Proceedings of the 17th international conference on machine learning* (pp. 743–750).

Prudêncio, R., & Ludermir, T. (2004). Meta-learning approaches to selecting time series models. *Neurocomputing*, *61*, 121–137.

Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, *15*, 65118.

Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754.

Rondn-Villarreal, P., Osorio, D., & Torres, R. (2014). *Peptides: Calculate indices and theoretical physicochemical properties of peptides and protein sequences*. http://CRAN.R-project.org/package=Peptides.

Sahigara, F., Mansouri, K., Ballabio, D., Mauri, A., Consonni, V., & Todeschini, R. (2012). Comparison of different approaches to define the applicability domain of QSAR models. *Molecules*, *17*(5), 4791–4810.

Segal, M., & Xiao, Y. (2011). Multivariate random forests. *Wiley interdisciplinary reviews: Data mining and knowledge discovery, 1*(1), 80–87. ISSN: 19424787. https://doi.org/10.1002/widm.12.

Smith-Miles, K. A. (2008). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR), 41*(1), 6:1–6:25.

Smith, M. R., Martinez, T. R., & Giraud-Carrier, C. G. (2014a). An instance level analysis of data complexity. *Machine Learning*, *95*(2), 225–256. https://doi.org/10.1007/s10994-013-5422-z.

Smith, M. R., Mitchell, L., Giraud-Carrier, C., Martinez, T. R. (2014b). Recommending learning algorithms and their associated hyperparameters. In *Proceedings of the international workshop on meta-learning and algorithm selection co-located with 21st European conference on artificial intelligence, MetaSel@ECAI 2014, Prague, Czech Republic, August 19, 2014* (pp. 39–40).

Soares, C., & Brazdil, P. (2000). Zoomed ranking: Selection of classification algorithms based on relevant performance information. In *Proceedings of the 4th European conference on principles of data mining and knowledge discovery (PKDD-2000)* (pp. 126–135).

Soares, C., Brazdil, P., & Kuba, P. (2004). A meta-learning method to select the kernel width in support vector regression. *Machine Learning*, *54*, 195–209.

Thornton, C., Hutter, F., Hoos. H. H, & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'13)*.

Todorovski, L., Blockeel, H., & Dzeroski, S. (2002). Ranking with predictive clustering trees. *Lecture Notes in Computer Science*, *2430*, 444–455.

van Rijn, J. N., Abdulrahman, S. M., Brazdil, P., & Vanschoren, J. (2015a). Fast algorithm selection using learning curves. In *Advances in intelligent data analysis XIV—14th international symposium, IDA 2015, Saint Etienne, France, October 22–24, 2015, Proceedings* (pp. 298–309).

van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2014). Algorithm selection on data streams. In *Discovery science—17th international conference, DS 2014, Bled, Slovenia, October 8–10, 2014. Proceedings* (pp. 325–336).

van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2015b) Having a blast: Meta-learning and heterogeneous ensembles for data streams. In *2015 IEEE international conference on data mining, ICDM 2015, Atlantic City, NJ, USA, November 14–17, 2015* (pp. 1003–1008).

Vanschoren, J. (2010). *Understanding learning performance with experiment databases*. Ph.D. Thesis. University of Leuven.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, *15*(2), 49–60. https://doi.org/10.1145/2641190.2641198.

Williams, A. J., Ekins, S., & Tkachenko, V. (2012). Towards a gold standard: Regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug Discovery Today*, *17*(13–14), 685–701.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques, Second Edition (Morgan Kaufmann series in data management systems)*. San Francisco, CA: Morgan Kaufmann Publishers Inc. ISBN: 0120884070.

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Xiao, N., Cao, D. S., Zhu, M. F., & Xu, Q. S. (2015). protr/protrweb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics*, *31*, 1857–1859. https://doi.org/10.1093/bioinformatics/btv042.

Xu, L., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2008). SATzilla: Portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, *32*, 565606.

Xu, L., Hutter, F., Shen, J., Hoos H. H., & Leyton-Brown, K. (2012). SATzilla2012: Improved algorithm selection based on cost-sensitive classification models. In *Proceedings of SAT Challenge 2012*.