JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Université Nice Sophia Antipolis

Subscriber access provided by Université Nice Sophia Antipolis

Computational Biochemistry

Stacked Generalization with Applicability Domain Outperforms simple QSAR on in vitro Toxicological Data

Ingrid Grenet, Kevin Merlo, Jean Paul Comet, Romain Tertiaux, David Rouquié, and Frederic Dayan

J. Chem. Inf. Model., **Just Accepted Manuscript** • DOI: 10.1021/acs.jcim.8b00553 • Publication Date (Web): 08 Feb 2019

Just Accepted

ACS Publications

is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

Stacked Generalization with Applicability Domain Outperforms simple QSAR on *in vitro* Toxicological Data

Ingrid Grenet,[†,‡,§] Kevin Merlo,[*,¶,§] Jean-Paul Comet,[†] Romain Tertiaux,[¶] David Rouquié,[‡] and Frédéric Dayan[¶]

University Côte d'Azur, I3S laboratory, UMR CNRS 7271, CS 40121, 06903 Sophia Antipolis Cedex France, Bayer SAS, 06903 Sophia Antipolis Cedex France, and Dassault Systèmes SE, 06906 Sophia Antipolis Biot France

E-mail: Kevin.MERLO@3ds.com

Abstract

The development of *in silico* tools able to predict bioactivity and toxicity of chemical substances is a powerful solution envisioned to assess toxicity as early as possible. To enable the development of such tools, the ToxCast program has generated and made publicly available *in vitro* bioactivity data for thousands of compounds. The goal of the present study is to characterize and explore the data from ToxCast in terms of Machine Learning capacity. For this, a large scale analysis on the entire database has been performed to build models to predict bioactivities measured in *in vitro* assays. Simple classical QSAR algorithms (ANN, SVM, LDA, Random Forest and Bayesian) were first

[*]To whom correspondence should be addressed
[†]I3S laboratory
[‡]Bayer SAS
[¶]Dassault Systemes
[§]Equal contributor

applied on the data, and the results of these algorithms suggested that they do not seem to be well suited for datasets with a high proportion of inactive compounds. The study then showed for the first time that the use of an ensemble method named "Stacked generalization" could improve the model performance on this type of data. Indeed, for 61% of 483 models, the Stacked method led to models with higher performance. Moreover, the combination of this ensemble method with an applicability domain filter allows one to assess the reliability of the predictions for further compound prioritization. In particular we showed that for 50% of the models, the ROC score is better if we do not consider the compounds that are not within the applicability domain.

Introduction

Chemical risk evaluation is aiming at defining the safe conditions of use of chemicals for both human health and the environment. To do so, the various hazards of the compounds are determined in a series of *in vitro* assays and *in vivo* studies using rodent and non-rodent laboratory animals. However, these *in vivo* studies are expensive with regard to time, money and animals, and are not adapted for the evaluation of thousands of chemicals. Therefore, alternative solutions were envisioned to assess, as early as possible, the potential toxicity of compounds and to prioritize them for further testing. The Tox21 partnership between the US Environmental Protection Agency (US EPA), the National Institutes of Health and the National Toxicology Program proposed to address in an eponymous project these issues by using high-throughput screening (HTS) *in vitro* assays to determine bioactivities of selected chemical substances.[1] This project was expected to help in characterizing pathways of toxicity based on chemical bioactivity profiles, in prioritizing compounds for further targeted toxicity testing, and in developing predictive models for toxicity.[2] The ToxCast program led by the EPA in response to the Tox21 suggestions is one of the initiatives which generated public data regarding bioactivity of more than 8000 compounds tested in up to 900 HTS assays.[3] The main goals of the ToxCast project were to identify *in vitro* assays that were

relevant to predict *in vivo* toxicity, to develop predictive models based on these assays, and to use these models to prioritize compounds for further screening and testing.[4] Since the beginning of ToxCast in 2007, several computational methods such as Machine Learning (ML) and other statistical analysis have been proposed in order to establish relationships between *in vitro* data and *in vivo* effects. For example, Judson[5] used a set of *in vitro* assays from ToxCast to develop classifiers of *in vivo* toxicity using different ML methods. He showed that some methods are more appropriate than others for classifying the available *in vivo* toxicity data, which was largely derived from marketed compounds which tend to be of low toxicity and thus highly imbalanced. Also, Martin[6] and Sipes[7] used Linear Discriminant Analysis algorithm to predict *in vivo* reproductive toxicology and developmental toxicology respectively, based on specific *in vitro* assays. Other approaches tried to link bioactivity to *in vivo* outcomes using statistical methods such as correlation,[4] unsupervised multivariate analysis[8] or linear additive model.[9,10] Finally, *in vitro* assays have been combined with chemical structures to predict *in vivo* toxicity in ML models.[11,12] Nonetheless, since HTS programs can also be costly and time-consuming, there is a real interest in developing *in silico* methods, in particular, Quantitative Structure-Activity Relationship (QSAR) models predicting compound activity based on the chemical structure information,[13] to limit the use of *in vitro* assays.[14,15] Such QSAR models are used by regulatory agencies for risk and safety assessment[16] and the Organization for Economic Cooperation and Development (OECD) proposed five principles to be met by a QSAR model to be used for regulatory purposes.[17] These principles require: (1) a defined endpoint, (2) an unambiguous algorithm, (3) a defined domain of applicability, (4) appropriate measures for model evaluation and (5) a mechanistic interpretation, if possible. Some QSAR models have already been developed to predict bioactivity measured in HTS assays for specific endpoints.[18,19] The Tox21 challenge in 2014 specifically highlighted the need for such *in silico* tools by looking for good models which predict the activation of nuclear receptor and stress response pathways.[14] In this paper we explore how ML can be applied to ToxCast data, and what the optimum uses of those data

are in order to build the most accurate classifiers based on chemical structural information. Since we did not want to focus on a specific endpoint, we performed a large scale analysis using all the available data which, to our knowledge, has never been done before. Indeed, several QSAR models were built to predict bioactivity measured in numerous assays using different types of molecular descriptors and different learning algorithms. As classical QSAR methods are sometimes not well suited for toxicity data, we evaluated if the combination of different algorithms proposed by ensemble techniques could lead to more robust models. Here we focus on the Stacked generalization technique[20] applied to toxicological data and show that it results in an improvement of the model performance. Also, in order to assess the relevance of the predictions the use of applicability domain (AD), as proposed by the OECD principle 3 was evaluated. To our knowledge, this is the first time that Stacked generalization method combined with applicability domain is applied to toxicological data.

Methods

ToxCast data

Both chemical structure and bioactivity data used in this work come from the ToxCast database and are publicly available on the US EPA website[1]. Overall, we retrieved 8599 unique substances, with information for each substance corresponding to a chemical name, molecular formula, CAS registry number (CASRN), and Simplified Molecular Input Line Entry Systems (SMILES) code.

Bioactivity data corresponding to the results of 1192 HTS *in vitro* assays was provided as a hit matrix with values of 0, 1, -1 and NA for each pair of compound/assay respectively meaning "inactive", "active", "undetermined" and "non assigned". A compound is determined as "active" in an assay if the half maximal activity concentration (AC50) could be measured.

[1]https://www.epa.gov/chemical-research/toxicity-forecaster-toxcasttm-data

Chemical descriptors computation

To build QSAR models, we computed chemical descriptors for all the compounds using their SMILES representation and two different software: PaDEL-Descriptor[21] which is open source, and Pipeline Pilot[22] developed by Dassault Systèmes BIOVIA. We computed 1D and 2D descriptors that respectively encode: (1) chemical composition such as in particular fragment counts and molecular formulae, and (2) topology determined from the molecular graph (number of rotable bounds, number of rings).

With PaDEL-Descriptor, we computed 1444 descriptors; in order to keep all of them and study their relative importance for ML, we chose to remove compounds for which at least one descriptor value could not be computed. With Pipeline Pilot, 164 descriptors were computed. This constitutes the first step of the data processing workflow, see Figure 1-A. The full lists of descriptors are available in supplementary materials.

From ToxCast raw data to processed datasets

We built one dataset for each ToxCast bioassay, characterized by molecular descriptors as input and the assay result as output. A total of 2384 datasets were generated (1192 with Pipeline Pilot descriptors and 1192 with PaDEL descriptors), corresponding to the 1192 ToxCast *in vitro* assays. For each dataset, we removed the compounds for which the assay value was not available (NA or -1) in order to build classifiers that can predict the positive or negative results of the assays Figure 1-A. As a consequence, 200 datasets containing only NA or -1 values were discarded. Finally, we ended up with 2184 datasets (1092 for each types of descriptors). Then, in order to discard irrelevant descriptors, we removed the ones that had a variance close to 0, and to limit redundancy in the dataset, we removed one of two correlated descriptors using a threshold of 0.8. Moreover, we kept only datasets that include at least 10 members of each class (i.e. 10 "active" compounds and 10 "inactive" ones) and that have at least 5 times more observations than descriptors. Finally, we ended up with 515 datasets with Pipeline Pilot descriptors and 414 datasets with PaDEL descriptors,

and, respectively denoted PLP datasets and PaDEL datasets (all the datasets are available in the supplementary material).

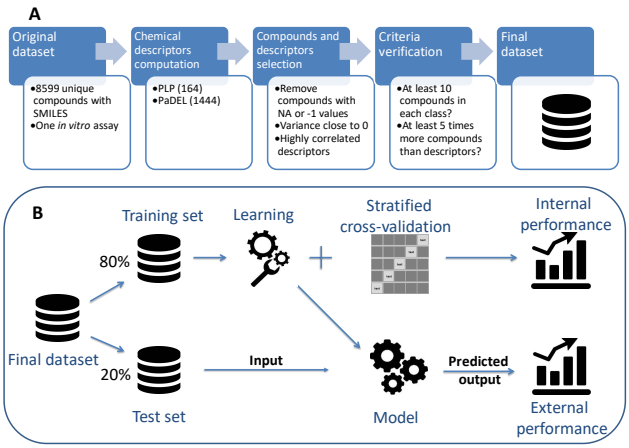

Figure 1: **Workflow of the data processing and learning procedure for one *in vitro* assay. A – Data processing:** For each *in vitro* assay, the original dataset contains the list of tested compounds with their structure from which chemical descriptors are computed. The compounds for which the reported value for the assay is NA or -1 are removed and the chemical descriptors are then selected according to their variance and correlation. Only datasets containing at least 10 compounds of the two classes (positive and negative) and at least 5 times more compounds than descriptors are kept. They correspond to the final datasets. **B – Learning procedure:** the dataset obtained in A is split into training set (80%) and test set (20%). The training set is used to learn the model using one of 5 different algorithms and a stratified 5-fold cross-validation is performed to get the internal performance of the model. The test set is then used to compute external performance of the model.

Simple QSAR classifiers

The learning procedure applied to each dataset is described in Figure 1-B.

We built ML models for all the datasets obtained previously using five supervised ML algorithms. Three algorithms (single hidden layer Artificial Neural Network (ANN), Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA)) were provided by the R packages NNET e1071, and MASS.[23–25] Two other algorithms (Forest multi-tree recursive partitioning classifier,[26] denoted hereafter as RPForest, and two-class Laplacian-modified Bayesian classifier,[27] denoted Bayesian) were implemented in Pipeline Pilot. For the 5 algorithms, we used the default parameters of the Pipeline Pilot machine learning collection.

We performed a stratified 5-fold internal cross-validation for each ML algorithm. To guarantee that each fold contained at least one member of each class of molecules, we first separated active and inactive molecules and the five folds were generated by randomly picking 20% from each class. For external validation, each dataset was randomly split into a training set (80%) and a test set (20%). The process was repeated 5 times and the average of the performance metrics was computed. Note that we chose to use random splitting by common use but other alternatives exist such as temporal split[28] or chemical clustering split.[29]

We used four metrics to evaluate the classification performance of the models, where T P (resp. TN) is the number of true positive (resp. negative), and F P (resp. FN) is the number of false positive (resp. negative): Sensitivity ($TP/(TP+FN)$), Specificity ($TN/(TN+FP)$), Balanced Accuracy, BA for short ($(Sensitivity + Specificity)/2$) and ROC score (area under Receiver Operating Characteristic curve) where ROC curve is the plot of $Sensitivity$ against $(1 - Specificity)$.

In order to perform 2-class predictions, the predicted continuous number that was returned by the algorithm was transformed into a binary one according to a threshold. There are several ways to determine this threshold and it has been shown that the traditional default method (threshold = 0.5) was unreliable for most of the datasets.[30] One of the best approach is to maximize the percentage of correctly classified observations. We chose an equivalent approach which is based on the minimization of the balanced error rate: $1\frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$.

Stacked Generalization

The Stacked generalization technique is an ensemble method which consists of the training of a learning algorithm that combines the predictions of several other algorithms. We randomly split each dataset into one training set (Train 1) and two test sets (Test 1 and Test 2) in the following proportions: 60%, 24% and 16% (see ovals in Figure 2). On Train 1, we trained a Bayesian and an RPForest algorithm to build models B1 and RP1 and we tested them on Test 1 to obtain predictions P1-b and P1-rp (see the blue workflow in Figure 2). We then

used these predictions P1-b and P1-rp as input descriptors with the outputs of Test 1 to train a so-called Stacked model. We chose a naïve Bayesian algorithm to build the final Stacked models (orange workflow of Figure 2) due to its performance and its ease of use in particular compared to Random Forest algorithm. We performed a 5-fold cross-validation repeated three times to find the best cutoff for the Stacked model. In some cases, it was impossible to compute a 5-fold cross-validation because of a small number of active compounds in Test 1: in these cases, the whole dataset was removed. Finally, we evaluated external predictive performance of the Stacked model on Test 2 (orange workflow of Figure 2). We applied this workflow for the 515 PLP datasets and generate 483 Stacked models. In order to compare the performance of the Stacked model with the simple QSAR classifiers, we merged Train 1 and Test 1 to train simple Bayesian and Random Forest learners and build models B2 and RF2 (see red workflow in Figure 2), and computed their external predictive performance on Test 2.

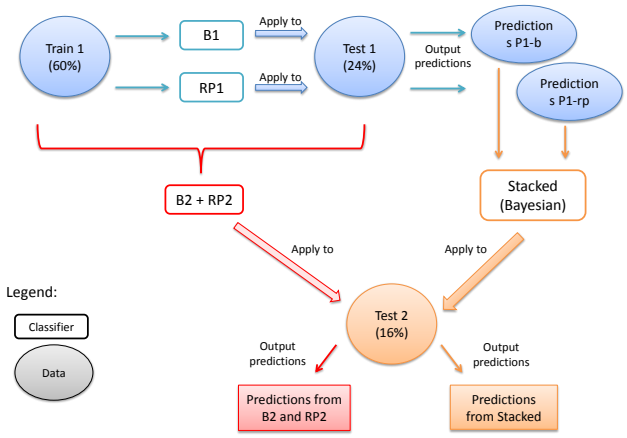

Figure 2: **Principle of the Stacked generalization method. Blue Workflow:** A Bayesian B1 and a RPForest RP1 models are built using Train 1 (60% of the dataset). The models are then used to make predictions P1-b and P1-rp on Test 1.
Orange workflow: the predictions P1-b and P1-rp are used to train a Stacked Bayesian model (Stacked).
Red workflow: By merging Train 1 and Test 1 to train a simple Bayesian model B2 and simple RPForest model RP2, we are able to compare performance of B2, RP2 and the Stacked model on Test 2.

Applicability Domain

We used two different approaches to compute the AD.

In the first approach, we performed a principal components analysis (PCA) directly applied on the chemical descriptors (1D and 2D) of the compounds used in the training set in order to reduce the space to only a few principal components (PCs). Basically, the PCs are computed to explain a minimum of 80% of the variance or a minimum of 10 components if 80% of the variance is explained with fewer components. Therefore, the number of principal components depends on the datasets. This allowed computation, for each PCA descriptor, of a range of acceptable values, as defined by the minimal and maximal PCA values observed in the training set. Then for a new compound, if the value of at least one of its PCA descriptors was out of the previously computed range, the compound was flagged "out of domain". For each model we computed the average performance obtained on an external test set using either all the compounds or only the compounds that are in the AD (note that when the test set had only one active compound in the AD, the dataset was removed, leaving only 487 datasets).

In the second approach, we computed the average Euclidean distance from each molecule of the test set to its three closest compounds from the training set.[31] After sorting the compounds in ascending order of the average distance, we first cut the test set into blocks of 50 compounds. We then counted the number of good predictions in each of these disjoint blocks.

Results and discussion

Datasets are imbalanced

The total number of compounds in the PLP datasets ranged between 1115 and 7810 with average and median values of 3054 and 3362 compounds respectively (Figure 3-A). Overall

58% of the datasets (300/515) had less than 10% active compounds (positive in the *in vitro* assay) (Figure 3-B). Regarding PaDEL datasets, their size varied from 1391 to 7516, and there was no dataset with more than 50% active compounds (data not shown). Moreover, 66% of the datasets (275/414) had less than 10% active compounds. Based on a binary classification of the assay values, the datasets were highly imbalanced in favor of inactive compounds (negative in the *in vitro* assays).

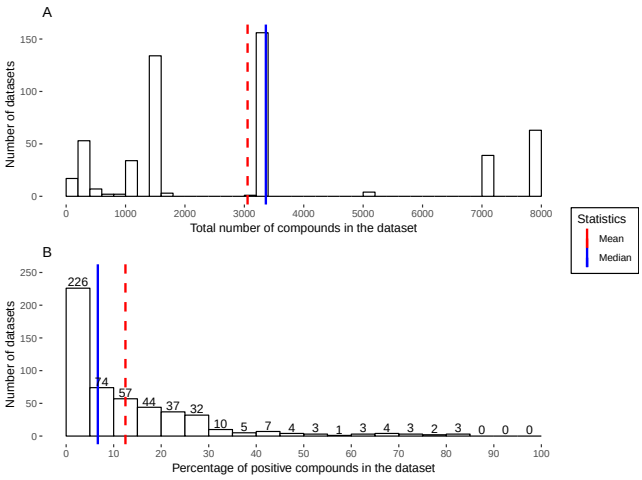

Figure 3: **A) Distribution of the 515 PLP datasets according to the number of compounds in the datasets. B) Distribution of the 515 PLP datasets according to the percentage of active compounds in the datasets.** 58% of the datasets (300/515) have less than 10% of active compounds. Similar observations are made for the PaDEL datasets (data not shown).

Simple QSAR models

Performance of 5 simple supervised learning methods

We trained five ML algorithms on the two types of datasets (PLP and PaDEL) and the mean of each performance metric (*Sensitivity*, *Specificity*, *ROCscore* and *BA*) was computed over the 515 and 414 datasets. Results are presented in Figure 4. For the two types of descriptors and the five algorithms, the four metric means were between 0.6 and 0.73 (except ANN-PaDEL *Sensitivity* which was at 0.52) and standard deviations were ranging between 0.06 and 0.19. Overall these results indicate that all algorithms performed similarly on this

type of data and have performance which are below our expectations.

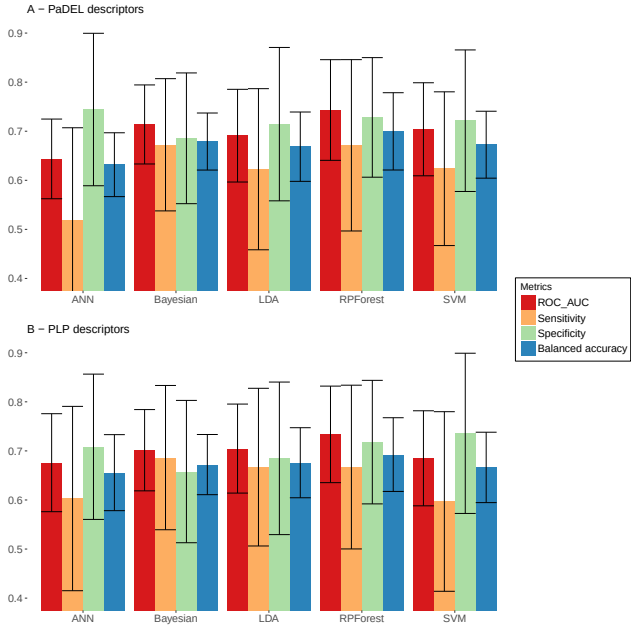

Figure 4: **Comparison of performance metrics (*ROCscore*, *Sensitivity*, *Specificity*and *BA*) after internal cross-validation for the 5 ML algorithms. A- Models using PaDEL descriptors and all datasets (414). B- Models using PLP descriptors and all datasets (515).**There is no difference between the 5 algorithms used, both for PaDEL and PLP.

As we already showed that the data are imbalanced, we considered if this characteristic can explain these results. For all algorithms, plots of BA obtained for each dataset according to the percentage of active compounds display a "funnel" shape with BA variability descreasing when datasets were more balanced (Figure 5). In particular, we observed lower BA variability for datasets containing at least 10% of compounds in the minority class. For datasets with low percentage of positives, most of the BA variability BA is due to the variability of the Sensitivity, which depends on the number of positive compounds in the datasets: for a same percentage of positive compounds, the larger the number of positives, the higher the Sensitivity. We obtained similar results with PaDEL datasets (data not shown).

Figure 6 presents the variance of *ROCscore* obtained with RPForest algorithm trained

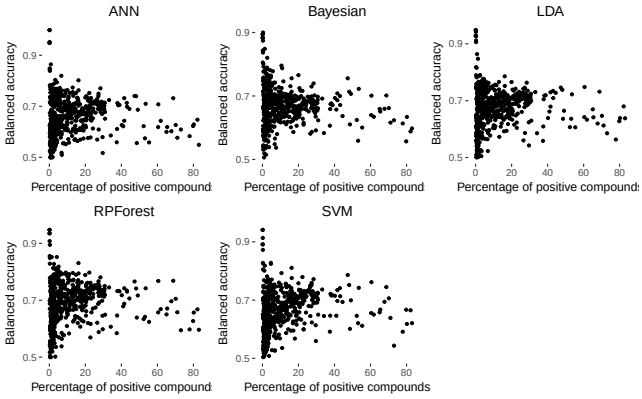

Figure 5: **Balanced accuracy according to percentage of positive compounds in PLP datasets for the 5 ML algorithms.** *BA* is getting stable when percentage of positive compounds in datasets increases. For datasets with low percentage of positives, most of the BA variability is due to the variability of the Sensitivity, which depends on the number of positive compounds in the datasets.

on PLP datasets according to the percentage of active compounds in the datasets. For each different range of percentage of positive compounds, we computed the variance of *ROCscore* over all the datasets having a percentage of positive compounds in that range. Figure 6 shows that the variance of the *ROCscore* tends to decrease when datasets are more balanced. We also see a cut-off: when datasets contain at least 10% of positive compounds, the associated variance is always below 0.0065. Similar results were obtained with PaDEL datasets (data not shown).

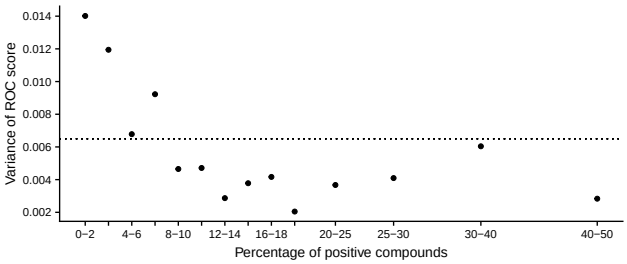

Figure 6: **Variance of *ROCscore* according to the percentage of positive compounds after internal cross-validation for Random Forest models based on PLP descriptors.** Variance is lower than 0.0065 when percentage of positive compounds is greater than 10%.

These results suggest that the imbalanced nature of datasets has a negative impact

on model performance and that these classical learning methods are not suitable for highly imbalanced datasets. This finding is in agreement with previous work from different domains showing that most classifier algorithms assume a relatively balanced distribution of the data.[32–35] Moreover, the models with very few positive compounds will be characterized by a limited applicability domain regarding this class of compounds. Consequently, a new positive compound will have a low chance to be within the applicability domain and its associated prediction will be of low confidence. In order to be able to follow principle 3 of the OECD recommandations about the use of QSAR modelling for regulatory purposes and based on the fact that our results are in line with previous published work, the datasets containing less than 10% of compounds that belong to the minority class were excluded in the following analysis.

Performance on more balanced datasets

When focusing on datasets containing at least 10% compounds in the minority class, we ended up with 139 PaDEL and 215 PLP datasets. Figure 7 shows the means of each performance metric over these datasets. The means of all metrics were between 0.63 and 0.75 and here again the results suggested that all the algorithms present similar performance (BA around 0.68). Interestingly, standard deviations were lower than previously: BA standard deviation decreased in average from 0.07 to 0.04, $Sensitivity$ from 0.16 to 0.10 and $Specificity$ from 0.14 to 0.09. In order to quantify the advantage of using more balanced datasets, we performed a Student test: Table 1 shows the p-values of the t.test that compared the mean of the 4 metrics between the datasets containing striclty less than 10% of compounds of the minority class and the ones containing at least 10% of compounds, for the 5 algorithms and the 2 types of descriptors used. Most of the metrics means are significantly different, meaning that the use of more balanced datasets has an impact on the models performance. In particular, the $Sensitivity$ is always significantly increased suggesting that the use of more balanced datasets, and therefore more positive compounds in that case, helps in

the detection of true positives. Overall, the same tendencies were observed with both types of descriptors but PLP datasets contain fewer descriptors which confers several advantages. They are generally easier to understand as they are related to well-known physico-chemical properties such as molecular weight or solubility. Also, other advantages of building ML models with fewer descriptors are decreased model complexity, reduced chances of overfitting,[36,37] and decreased computational time. Therefore, we made the decision to focus only on PLP datasets for the rest of the study.

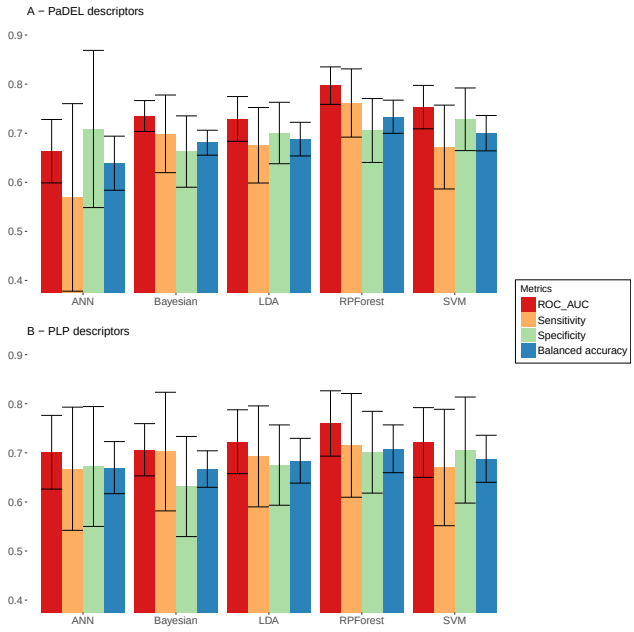

Figure 7: **Comparison of performance metrics (*ROCscore*, *Sensitivity*, *Specificity* and BA) after internal cross-validation for the 5 ML algorithms. A- Models using PaDEL descriptors and datasets with at least 10 percent of compounds in the minority class (139). B- Models using PLP descriptors and datasets with at least 10% of compounds in the minority class (215).** There is no difference between the 5 algorithms used, both for PaDEL and PLP. Standard deviations get smaller when we keep datasets with at least 10% of compounds in the minority class.

External validation

Figure 8 presents the average *BA*, *Sensitivity* and *Specificity* obtained after external validation for the 5 ML algorithms on all 215 PLP datasets. We observed that, except for the Bayesian algorithm, *Sensitivity* was very low (under 0.4) and *Specificity* was high (greater

Table 1: **p-values of Student test performed on the 4 metrics for the 5 algorithms and 2 types of descriptors, between the datasets that contain strictly less than than 10% of active compounds and the datasets that contain at least 10% of active compounds. The p-values lower than 0.05 are in bold.** Most of the metrics means are significatively different meaning that the use of more balanced datasets has an impact on the models performance.

Method	Descriptor type	ROC AUC	Balanced Accuracy	Sensitivity	Specificity
ANN	PADEL	$\mathbf{1.10 \times 10^{-4}}$	9.06×10^{-2}	$\mathbf{1.58 \times 10^{-4}}$	$\mathbf{1.15 \times 10^{-3}}$
Bayesian	PADEL	$\mathbf{6.32 \times 10^{-7}}$	5.96×10^{-1}	$\mathbf{5.95 \times 10^{-4}}$	$\mathbf{2.00 \times 10^{-3}}$
LDA	PADEL	$\mathbf{1.29 \times 10^{-13}}$	$\mathbf{4.85 \times 10^{-7}}$	$\mathbf{2.98 \times 10^{-9}}$	9.03×10^{-2}
RPForest	PADEL	$\mathbf{2.40 \times 10^{-23}}$	$\mathbf{1.04 \times 10^{-15}}$	$\mathbf{1.70 \times 10^{-22}}$	$\mathbf{8.48 \times 10^{-4}}$
SVM	PADEL	$\mathbf{8.76 \times 10^{-22}}$	$\mathbf{3.55 \times 10^{-13}}$	$\mathbf{3.85 \times 10^{-8}}$	3.73×10^{-1}
ANN	PLP	$\mathbf{1.66 \times 10^{-7}}$	$\mathbf{1.20 \times 10^{-4}}$	$\mathbf{3.30 \times 10^{-13}}$	$\mathbf{7.59 \times 10^{-7}}$
Bayesian	PLP	2.11×10^{-1}	7.82×10^{-2}	$\mathbf{2.55 \times 10^{-2}}$	$\mathbf{1.47 \times 10^{-4}}$
LDA	PLP	$\mathbf{3.79 \times 10^{-5}}$	$\mathbf{1.69 \times 10^{-2}}$	$\mathbf{6.80 \times 10^{-4}}$	1.79×10^{-1}
RPForest	PLP	$\mathbf{2.15 \times 10^{-8}}$	$\mathbf{9.39 \times 10^{-6}}$	$\mathbf{8.98 \times 10^{-10}}$	$\mathbf{5.69 \times 10^{-3}}$
SVM	PLP	$\mathbf{6.71 \times 10^{-15}}$	$\mathbf{2.31 \times 10^{-10}}$	$\mathbf{1.46 \times 10^{-17}}$	$\mathbf{1.09 \times 10^{-4}}$

than 0.8) which led to a BA around 0.6. Moreover, standard deviations of data obtained with ANN, LDA, RPForest and SVM algorithms were large, and we were unable to draw conclusions on the usefulness of this method based on these results. These external performance make us think that the 4 ML methods are not able to build good models, even when focusing only on datasets having at least 10% of compounds in the minority class. We hypothesized that they were too sensitive to imbalanced datasets as Mazurowski[38] showed for neural networks. Indeed, Mazurowski performed a large scale analysis on simulated imbalanced data and studied the impact of this characteristic on two neural network methods (classical backpropagation and particle swarm optimization). He concluded that even a small imbalance in the training set leads to a deterioration of the performance.

On the contrary the Bayesian algorithm seems to be more suited to imbalanced datasets since BA, $Sensitivity$ and $Specificity$ are greater than 0.6 with smaller standard deviations.

Overall, our results suggest that a unique algorithm alone was not able to generate models with sufficient performance.

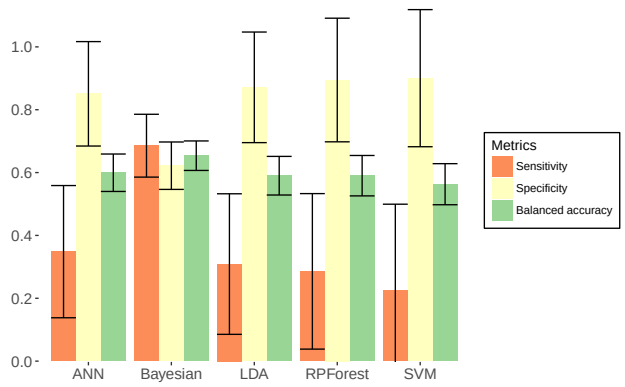

Figure 8: **Comparison of performance metrics (sensitivity, *Specificity* and balanced accuracy) after external validation for the 5 algorithms on PLP datasets with at least 10% of compounds in the minority class (215 datasets).** ANN, LDA, RPForest and SVM are not able to build good models whereas Bayesian algorithm seems to be more suitable.

An ensemble technique: the Stacked generalization

The next step was to test if combining different types of algorithms would improve the results. This approach has been widely studied and led to different approaches so-called "ensemble techniques".[39] Intuitively, each learning algorithm makes a hypothesis to predict as well as possible a particular output. When the choice of these algorithms (and the underlying hypothesis) is not obvious, taking an "ensemble" of models trained on a same dataset allows the combination of multiple hypotheses in a unique model leading to higher performance.[39,40] Here, we used the Stacked generalization technique[20,41] using two methods based on very different internal representations of the training sets (instances and trees): Bayesian and RPForest algorithms which have been shown to lead to the best *Sensitivity* (Bayesian) and high *Specificity* (RPForest). The final stacked model was built using a naïve Bayes classifier. Note that we also built stacked models using 3 to 5 base models (by iteratively adding ANN, SVM and LDA to RPForest and Bayesian) on a subset of datasets (data not shown). Since the performance did not change significantly according to the number of models used, we therefore decided to keep only two base models for a matter of computing time.

Stacked generalization vs. simple QSAR classifiers

We compared the predictive performance of the 483 simple Bayesian and RPForest models (B2 and RP2) with the Stacked ones. Table 2 shows, for the three types of methods, the number of models that reach a certain value of *ROCscore*. First, less RP2 models are able to reach *ROCscore* greater than 0.6 compared to the B2 and Stacked ones and only 30 among the 483 have *ROCscore* above 0.8. Furthermore, if an equivalent number of B2 and Stacked models reach 0.6 and 0.7 values of *ROCscore*, when looking at higher and better performance (above 0.75 and 0.8), the Stacked method becomes clearly better than the Bayesian one. Finally, when we look at the method that lead to the highest *ROCscore* for each model, the Stacked is the winner for 61% models (294/483), followed by the Bayesian one with 30% of models (147/483) and the RPForest with only 9% (42/483) (data not shown). These results confirm that, also in our application and based on the *ROCscore* values, the Stacked generalization method is able to build more models with good performance than simple QSAR algorithms.

Table 2: **Comparison of simple Bayesian B2 and Random Forest RP2 models with Stacked generalization models on the 483 PLP datasets.** The comparison of the number of models that reach a certain value of *ROCscore* shows that the Stacked generalization method is able to build more models with higher *ROCscore* than simple QSAR methods.

Method	*ROC* $\geq$ 0.60	*ROC* $\geq$ 0.70	*ROC* $\geq$ 0.75	*ROC* $\geq$ 0.80
Stacked	417	319	253	144
Bayesian	416	321	223	90
RPForest	356	205	89	30

Figure 9 shows the ROC curves of B2 and stacked models for one particular assay (TOX21_ERa_BLA_Antagonist_ratio) measuring the expression of the Estrogen Receptor gene[2] and for which model performance were among the best. The associated dataset contains 7810 molecules, 13% of which were active in the *in vitro* assay. We observed that the ROC curve of the Stacked model is always above that of the B2 model. For this particular

[2]see `https://actor.epa.gov/dashboard`.

assay, the *ROCscores* were 0.84 and 0.77 for Stacked and B2 models respectively. Furthermore, since the ROC curve displays the *Sensitivities* and their corresponding *Specificities* obtained for all threshold values between 0 and 1, we can choose a specific threshold according to a desired *Sensitivity* or *Specificity*. As an example, a *Sensitivity* of 85% corresponds to a *Specificity* of about 73% with the Stacked model and only of 52% with the Bayesian one, see dotted lines in Figure 9. This again illustrates the difference of performance between the two models and the ability of the Stacked model to detect more inactive compounds than the Bayesian one, for the same number of active detected. Naturally, one can move this threshold depending on the necessary stringency of the model output.

The same analysis on other assays led to the same observations and conclusions (data not shown).

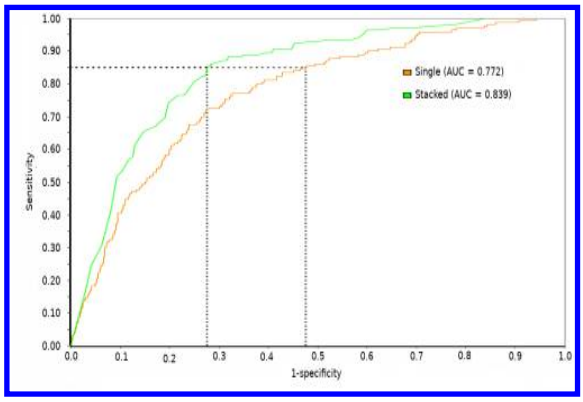

Figure 9: **ROC curves obtained for models trained on the dataset of the assay TOX21_ERa_BLA_Antagonist_ratio with the two types of methods (Stacked generalization and simple Bayesian).** ROC curve of the Stacked model is always above the one of simple Bayesian model. For a given *Sensitivity* of 85%, the Bayesian model detects 52% of the inactive molecules whereas the Stacked model detects 73% of them.

Focus on *in vitro* bioactivity assays proved to be linked to *in vivo* toxicity

Since the bioactivity assays can be seen as an intermediate step towards the evaluation of the *in vivo* toxicity, several works have relevantly focused on the link between ToxCast *in vitro* assays and toxicity outcomes observed *in vivo*. In particular, in 2015 Liu and co-workers built ML models that predict *in vivo* chronic toxicity observed in liver[12] based on either

chemical descriptors, bioactivity descriptors (i.e ToxCast *in vitro* assays) or a combination of both. This study has been extended in 2017 to 19 other organs.[42] They extracted in both studies the 36 (resp. 50) *in vitro* assays most frequently used in their models and which were supposed to be the most correlated with *in vivo* liver (resp. 19 other organs) toxicity.

Since we built QSAR models for the majority of the ToxCast *in vitro* assays, we proposed here to focus on the ones that predict these assays. More precisely, among the 36 (resp. 50) *in vitro* assays highlighted by Liu, we were able to build 25 (resp. 38) corresponding QSAR models. Because 11 models were common to both sets (25 and 38), we finally got 52 QSAR models, using the simple algorithms and the stacked method. Table 3 summarizes the best $ROCscore$ we obtained for these 52 QSAR models and the corresponding method used (simple Bayesian, RPForest or Stacked generalization). For 71% of the assays (37/52), the Stacked generalization was the method leading to the best ROC score. Also, for 62% of the assays (32/52) the $ROCscore$ was greater than 0.75 meaning that we were able to build good QSAR models to predict some of the *in vitro* assays highlighted by Liu. Altogether, these results show that the Stacked generalization method allows one to build QSAR classifier models that predict *in vitro* assays which have been previously shown to be associated to *in vivo* toxicity outcomes. This suggest that we could think about replacing all or part of these *in vitro* assays by *in silico* predictions and use these predictions as input of Liu's ML models for example.

Applicability domain information to reinforce the confidence of predictions

The OECD QSAR Validation principles recommend that a model should be used within its applicability domain (AD).[17] Several definitions of the AD have been proposed and the Setubal Workshop report[43] proposed the following one: "The AD of a (Q)SAR is the physicochemical, structural, or biological space, knowledge or information on which the training set of the model has been developed, and for which it is applicable to make predictions for

Table 3: **Most frequently used assays in Liu's models and the best *ROCscores* we obtained with either Stacked generalization or simple methods (Bayesian and RPForest). Assays are sorted by decreasing *ROCscore*.** Stacked generalization method has the best *ROCscore* for 71% of the assays and this score is greater than 0.75 for 62%.

Assay name	**Liver**	**Others**	**ROC** score score	**Bayesian(B)/ Stacked(S) RPForest (RP))**
BSK_KF3CT_SRB_down		×	0,855	S
TOX21_TR_LUC_GH3_Antagonist	×		0,842	S
APR_HepG2_CellLoss_24h_dn		×	0,842	B
TOX21_ERa_BLA_Antagonist_ratio	×		0,839	S
ATG_VDRE_CIS_up	×	×	0,836	S
ATG_SREBP_CIS_up		×	0,834	S
ATG_PBREM_CIS_up		×	0,834	S
ATG_MRE_CIS_up		×	0,834	S
ATG_PXR_TRANS_up		×	0,833	S
APR_HepG2_MitoticArrest_72h_up	×	×	0,830	S
TOX21_ERa_LUC_BG1_Antagonist		×	0,828	S
TOX21_PPARd_BLA_agonist_ratio		×	0,827	S
TOX21_Aromatase_Inhibition	×		0,821	S
ATG_TGFb_CIS_up		×	0,819	S
ATG_RARa_TRANS_up	×		0,815	S
APR_HepG2_StressKinase_1h_up		×	0,815	B
ATG_RORE_CIS_up		×	0,812	B
BSK_3C_Vis_down		×	0,811	S
ATG_PXRE_CIS_up	×	×	0,810	S
BSK_BE3C_SRB_down		×	0,809	S
ATG_NRF2_ARE_CIS_up	×		0,793	S
ATG_PPRE_CIS_up	×	×	0,791	S
NVS_GPCR_hOpiate_mu	×		0,791	S
ATG_RXRb_TRANS_up	×		0,787	S
ATG_C_EBP_CIS_up		×	0,785	S
ATG_LXRb_TRANS_up		×	0,780	B
ATG_BRE_CIS_up	×	×	0,780	S
TOX21_PPARg_BLA_antagonist_ratio		×	0,775	S
ATG_Oct_MLP_CIS_up	×		0,771	S
APR_HepG2_MicrotubuleCSK_72h_up		×	0,760	B
APR_HepG2_MitoMembPot_1h_dn		×	0,759	B
ATG_NFI_CIS_up		×	0,754	B
ATG_NF_kB_CIS_up		×	0,749	B
NVS_MP_rPBR	×	×	0,749	RP
NVS_ADME_hCYP2C19	×	×	0,733	S
APR_HepG2_CellCycleArrest_24h_up		×	0,733	B
NVS_NR_hAR	×		0,727	B
ATG_ERE_CIS_up		×	0,717	S
NVS_ADME_hCYP1A2	×		0,717	B
NVS_NR_mERa	×		0,708	S
ATG_IR1_CIS_up		×	0,708	S
APR_HepG2_CellCycleArrest_72h_dn		×	0,705	S
NVS_NR_hPXR	×		0,691	B
NVS_NR_hCAR_Antagonist		×	0,689	S
NVS_MP_hPBR	×	×	0,682	S
TOX21_ERa_LUC_BG1_Agonist	×	×	0,676	S
NVS_TR_hNET	×		0,668	S
APR_HepG2_NuclearSize_72h_up	×		0,650	B
NVS_NR_hER	×	×	0,649	S
TOX21_TR_LUC_GH3_Agonist		×	0,646	S
APR_HepG2_MitoMass_24h_up		×	0,599	B
APR_HepG2_MitoMass_72h_up	×	×	0,537	S

new compounds. [...] Ideally, the (Q)SAR should only be used to make predictions within that domain by interpolation not extrapolation". Basically, the AD enable to estimate the similarity between training set and test sets.[31,44] In practice, there are different approaches to estimate if a molecule is within the AD or out of it. We chose to consider two of these approaches, first the position of the new molecule in the space described by descriptors and second the distance between the new molecule and the closest molecules of the training set. Table 4 presents the results of the first approach and shows the percentage of assays for

which performance metrics were higher when considering the test set with the molecules in AD only compared to the entire test set. For more than 50% of the assays, *ROCscore* and *BA* were higher when "out of AD" compounds were excluded. *Specificity* was higher for 88% of the assays but only 16% showed higher *Sensitivity* when using only "in AD" compounds. This could be explained by the low number of active compounds in the test sets leading to equivalent performance regardless of the molecules taken into consideration. Moreover, the number of compounds "out of AD" was very low for all test sets (between 5 and 10) which led to similar test sets and also explains the results of Table 4. It could be interesting to test the importance of this AD approach for test sets with more "out of AD" compounds or to calculate the *BA* of compounds that are "out of AD".

Table 4: **Percentage of assays with higher performance when using only "in AD" compounds than when using all compounds.**

Metric	% of assays
ROC score	53.2
BA	53.0
Sensitivity	16.0
Specificity	88.3

We applied the second approach to the dataset of the assay TOX21_ERa_BLA_Antagonist_ratio which was reported by Liu to be linked to chronic liver toxicity.[12] The test set was composed of 1251 molecules, corresponding to 24 disjoint blocks of 50 molecules and one block of 51. Figure 10 displays the results of the AD analysis obtained for these 25 blocks for the simple Bayesian and Stacked models for which *ROCscores* are 0.77 and 0.84 respectively. We observed that the percentage of good predictions decreased (from 100% to less than 50%) when the average distance to the compounds in the training set increased. This shows that we can be more confident in the predictions when the compounds are close to the ones of the training set. Moreover, the percentage of good predictions with the Stacked model was greater than with the Bayesian one (except for blocks 20 to 23). We also performed the analysis for the assay TOX21_AR_BLA_Antagonist_ratio and found comparable results

(data not shown).

This analysis shows that the use of the Stacked generalization method to predict *in vitro* assays from the chemical structure, in association with a tool to measure whether a new compound belongs to the AD, can lead to good and reliable predictions when this new compund is actually inside the AD.

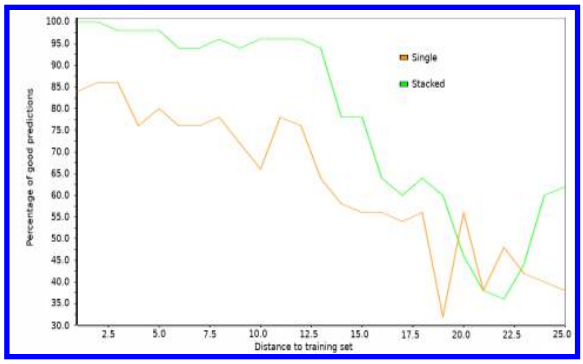

Figure 10: **Number of good predictions made by the Stacked model and the simple Bayesian in all blocks of 50 predictions according to the average distance between the molecules of the test set and the three closest compounds of the training set. The dataset used corresponds to the assay TOX21_ERa_BLA_Antagonist_ratio.** Percentage of good predictions decreases when the average Euclidean distance to the training set increases. Percentage of good predictions of Stacked model is almost always greater than that of the Bayesian model.

Conclusion

In this work, we performed a large scale analysis in order to characterize the Toxcast data which is generally very imbalanced with only a few active compounds for each *in vitro* assay. We first used classical learning methods, using two types of descriptors (PLP and PaDEL), in order to build models aiming at the prediction of results of ToxCast *in vitro* assays from chemical strucutures. The results indicated that all algorithms performed similarly and appeared to be below our expactations for this type of data. We then built Stacked models and, as recently shown by Madasamy,[45] demonstrated that this technique is more appropriate for this type of imbalanced data. In particular, we were able to build models

with good performance for *in vitro* assays that have previously been shown to be related to specific *in vivo* toxicity outcomes.[12,42]

Moreover, we demonstrated that the AD information is an important parameter to evaluate the reliability of predictions and can help to support decision making and prioritization. Indeed, it could be a complementary and stringent filter allowing the selection of compounds for further *in vivo* testing. For example, molecules within the AD and predicted as toxic would be the first to be further characterized *in vivo*. For Toxcast data, combining Stacked generalization methods with an AD filter led to better classifier than did the classical learning method for Toxcast data.

Choices made in this study highlight directions for future work. First, instead of building classifiers, one could use regression algorithms to predict the AC50 values obtained in each *in vitro* assay and not simply the "active" vs "inactive" labels. Here, by applying the threshold before the learning, we might have lost information. It could be interesting to see if more accurate models can be built by applying the threshold after the predictions. Second, we did not explore all the possible parameters for the different learners; these parameters can be tuned in order to increase the performance but the operation is time consuming.[46,47] Regarding the stacked generalization method, we could think about exploring more combinations of customized base models and using other algorithms as meta-learner to build the stacked model in order to reach higher performance. Third, to complement the 1D and 2D molecular descriptors used in this paper, we could also use 3D descriptors and fingerprints which are known to be good for QSAR.[48,49] Then, to face the effect of the highly imbalanced data, we could implement data augmentation techniques[50] which balance the training set by adding or removing data and are supposed to increase model performance.[51] Finally, we could test other ensemble techniques such as bagging, boosting or bucket of models[45] to compare with the stacked generalization method. We actually already tried to use bagging with Random Forest and Bayesian algorithms on 15 datasets (having more than 1000 compounds and at least 30% of positive ones) and these first results showed that the bagging did

not lead to better performance than the stacked generalization. However, this work needs to be extended to all the datasets and more methods. More interestingly we would like to evaluate the ability of predicting *in vivo* toxicity by considering both structure and *in vitro* information, either by combining them as input descriptors or by chaining up two prediction steps.

Acknowledgement

The authors thank Gabriel Sarrazin and Moeka Shishido who provided insights and comments that greatly improved the manuscript. The authors would also like to show their gratitude to the Dassault Systèmes BIOVIA Drug Design Services team members (Celine Ferre, Tanguy Devos and Patrice Pistone) for their support. Editing of the manuscript by Elizabeth Shipp and Leo Bleicher was gratefully appreciated by the authors.

This work was partly supported by the OSEO BioIntelligence program. I.G hold a doctoral fellowhip from the Association Nationale de la Recherche Technique (ANRT CIFRE PhD funding).

Finally, the authors acknowledge the reviewers for their comments and questions which have allowed us to improve the quality of the paper.

Notes The authors declare no conflict of interest.

Supporting Information Available

The following files are available free of charge.

- supplementary1_PaDELdescriptors.xlsx: list of PaDEL descriptors
- supplementary2_PLPdescriptors.xlsx: list of PLP descriptors
- supplementary3_datasets.zip: datasets used for learning

This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Council, N. R. *Toxicity Testing in the 21st Century*; National Academies Press: Washington, D.C., 2007.

(2) Knudsen, T.; Martin, M.; Chandler, K.; Kleinstreuer, N.; Judson, R.; Sipes, N. Predictive models and computational toxicology. *Methods in molecular biology (Clifton, N.J.)* **2013**, *947*, 343–374.

(3) Dix, D. J.; Houck, K. A.; Martin, M. T.; Richard, A. M.; Setzer, R. W.; Kavlock, R. J. The ToxCast Program for Prioritizing Toxicity Testing of Environmental Chemicals. *Toxicol. Sci.* **2007**, *95*, 5–12.

(4) Judson, R. S.; Houck, K. A.; Kavlock, R. J.; Knudsen, T. B.; Martin, M. T.; Mortensen, H. M.; Reif, D. M.; Rotroff, D. M.; Shah, I.; Richard, A. M.; Dix, D. J. In Vitro Screening of Environmental Chemicals for Targeted Testing Prioritization: The ToxCast Project. *Environ. Health Perspect.* **2009**, *118*, 485–492.

(5) Judson, R.; Elloumi, F.; Setzer, R. W.; Li, Z.; Shah, I. A comparison of machine learning algorithms for chemical toxicity classification using a simulated multi-scale data model. *BMC Bioinf.* **2008**, *9*, 241.

(6) Martin, M. T.; Knudsen, T. B.; Reif, D. M.; Houck, K. A.; Judson, R. S.; Kavlock, R. J.; Dix, D. J. Predictive Model of Rat Reproductive Toxicity from ToxCast High Throughput Screening1. *Biol. Reprod.* **2011**, *85*, 327–339.

(7) Sipes, N. S.; Martin, M. T.; Reif, D. M.; Kleinstreuer, N. C.; Judson, R. S.; Singh, A. V.; Chandler, K. J.; Dix, D. J.; Kavlock, R. J.; Knudsen, T. B. Predictive Models of Prenatal Developmental Toxicity from ToxCast High-Throughput Screening Data. *Toxicol. Sci.* **2011**, *124*, 109–127.

(8) Shah, I.; Houck, K.; Judson, R. S.; Kavlock, R. J.; Martin, M. T.; Reif, D. M.;

Wambaugh, J.; Dix, D. J. Using nuclear receptor activity to stratify hepatocarcinogens. *PloS one* **2011**, *6*, 1–11.

(9) Browne, P.; Judson, R. S.; Casey, W. M.; Kleinstreuer, N. C.; Thomas, R. S. Screening Chemicals for Estrogen Receptor Bioactivity Using a Computational Model. *Environ. Sci. Technol.* **2015**, *49*, 8804–8814.

(10) Kleinstreuer, N. C.; Ceger, P.; Watt, E. D.; Martin, M.; Houck, K.; Browne, P.; Thomas, R. S.; Casey, W. M.; Dix, D. J.; Allen, D.; Sakamuru, S.; Xia, M.; Huang, R.; Judson, R. Development and Validation of a Computational Model for Androgen Receptor Activity. *Chem. Res. Toxicol.* **2017**, *30*, 946–964.

(11) Thomas, R. S.; Black, M. B.; Li, L.; Healy, E.; Chu, T.-M.; Bao, W.; Andersen, M. E.; Wolfinger, R. D. A comprehensive statistical analysis of predicting in vivo hazard using high-throughput in vitro screening. *Toxicol. Sci.* **2012**, *128*, 398–417.

(12) Liu, J.; Mansouri, K.; Judson, R. S.; Martin, M. T.; Hong, H.; Chen, M.; Xu, X.; Thomas, R. S.; Shah, I. Predicting Hepatotoxicity Using ToxCast in Vitro Bioactivity and Chemical Structure. *Chem. Res. Toxicol.* **2015**, *28*, 738–751.

(13) Hansch, C. Quantitative structure-activity relationships and the unnamed science. *Acc. Chem. Res.* **1993**, *26*, 147–153.

(14) Capuzzi, S. J.; Politi, R.; Isayev, O.; Farag, S.; Tropsha, A. QSAR Modeling of Tox21 Challenge Stress Response and Nuclear Receptor Signaling Toxicity Assays. *Front. Environ. Sci.* **2016**, *4*, 3.

(15) Gadaleta, D.; Manganelli, S.; Roncaglioni, A.; Toma, C.; Benfenati, E.; Mombelli, E. QSAR Modeling of ToxCast Assays Relevant to the Molecular Initiating Events of AOPs Leading to Hepatic Steatosis. *J. Chem. Inf. Model* **2018**, *58*, 1501–1517.

(16) Roy, K.; Kar, S.; Ambure, P. On a simple approach for determining applicability domain of QSAR models. *Chemom. Intell. Lab. Syst.* **2015**, *145*, 22–29.

(17) OCDE, *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models*; OECD Series on Testing and Assessment; 2014; p 154.

(18) Ng, H. W.; Doughty, S. W.; Luo, H.; Ye, H.; Ge, W.; Tong, W.; Hong, H. Development and Validation of Decision Forest Model for Estrogen Receptor Binding Prediction of Chemicals Using Large Data Sets. *Chem. Res. Toxicol.* **2015**, *28*, 2343–2351.

(19) Zang, Q.; Rotroff, D. M.; Judson, R. S. Binary Classification of a Large Collection of Environmental Chemicals from Estrogen Receptor Assays by Quantitative Structure–Activity Relationship and Machine Learning Methods. *J. Chem. Inf. Model.* **2013**, *53*, 3244–3261.

(20) Wolpert, D. H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259.

(21) Yap, C. W. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *J. Comput. Chem.* **2011**, *32*, 1466–1474.

(22) Dassault Systèmes BIOVIA, *Pipeline Pilot v. 17.2.0.1361*; 2012.

(23) R Core Team, R: A Language and Environment for Statistical Computing. 2013; ISBN 3-900051-07-0.

(24) Meyer, D. Support Vector Machines. The Interface to libsvm in package e1071. 2001.

(25) Venables, W. N.; Ripley, B. D. *Modern Applied Statistics with S*, 4th ed.; Springer: New York, 2002.

(26) Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32.

(27) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *J. Med. Chem.* **2004**, *47*, 4463–4470.

(28) Sheridan, R. P. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model* **2013**, *53*, 783–790.

(29) Martin, T. M.; Harten, P.; Young, D. M.; Muratov, E. N.; Golbraikh, A.; Zhu, H.; Tropsha, A. Does Rational Selection of Training and Test Sets Improve the Outcome of QSAR Modeling? *J. Chem. Inf. Model* **2012**, *52*, 2570–2578.

(30) Freeman, E. A.; Moisen, G. G. A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecol. Modell.* **2008**, *217*, 48–58.

(31) Sheridan, R. P.; Feuston, B. P.; Maiorov, V. N.; Kearsley, S. K. Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1912–1928.

(32) Chawla, N. V.; Japkowicz, N.; Kotcz, A. Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter* **2004**, *6*, 1–6.

(33) Fawcett, T.; Provost, F. Adaptive fraud detection. *Data Min. Knowl. Discov* **1997**, *1*, 291–316.

(34) Kubat, M.; Holte, R. C.; Matwin, S. Machine learning for the detection of oil spills in satellite radar images. *Machine learning* **1998**, *30*, 195–215.

(35) Riddle, P.; Segal, R.; Etzioni, O. Representation design and brute-force induction in a Boeing manufacturing domain. *Applied Artificial Intelligence* **1994**, *8*, 125–147.

(36) Goodarzi, M.; Dejaegher, B.; Heyden, Y. V. Feature Selection Methods in QSAR Studies. *J. AOAC Int.* **2012**, *95*, 636–651.

(37) Guyon, I.; Elisseeff, A.; De, A. M. An Introduction to Variable and Feature Selection. *JMLR* **2003**, *3*, 1157–1182.

(38) Mazurowski, M. A.; Habas, P. A.; Zurada, J. M.; Lo, J. Y.; Baker, J. A.; Tourassi, G. D. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Netw.* **2008**, *21*, 427–436.

(39) Yang, P.; Hwa Yang, Y.; B. Zhou, B.; Y. Zomaya, A. A Review of Ensemble Methods in Bioinformatics. *Curr. Bioinf.* **2010**, *5*, 296–308.

(40) Zhang, L.; Ai, H.; Chen, W.; Yin, Z.; Hu, H.; Zhu, J.; Zhao, J.; Zhao, Q.; Liu, H. CarcinoPred-EL: Novel models for predicting the carcinogenicity of chemicals using molecular fingerprints and ensemble learning methods. *Sci. Rep.* **2017**, *7*, 2118.

(41) Güneş, F.; Wolfinger, R.; yi Tan, P. Stacked Ensemble Models for Improved Prediction Accuracy. SAS Global Forum Proc. 2017.

(42) Liu, J.; Patlewicz, G.; Williams, A. J.; Thomas, R. S.; Shah, I. Predicting Organ Toxicity Using in Vitro Bioactivity Data and Chemical Structure. *Chem. Res. Toxicol.* **2017**, *30*, 2046–2059.

(43) Roy, K.; Kar, S.; Das, R. N. *Understanding the basics of QSAR for applications in pharmaceutical sciences and risk assessment*; 2015.

(44) Sheridan, R. P. The Relative Importance of Domain Applicability Metrics for Estimating Prediction Errors in QSAR Varies with Training Set Diversity. *J. Chem. Inf. Model.* **2015**, *55*, 1098–1107.

(45) Madasamy, K.; Ramaswami, M. Data Imbalance and Classifiers: Impact and Solutions from a Big Data Perspective. *IJCIR* **2017**, *13*, 2267–2281.

(46) Snoek, J.; Larochelle, H.; Adams, R. P. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; 2012; pp 2951–2959.

(47) Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *JMLR* **2012**, *13*, 281–305.

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

(49) Zang, Q.; Mansouri, K.; Williams, A. J.; Judson, R. S.; Allen, D. G.; Casey, W. M.; Kleinstreuer, N. C. In Silico Prediction of Physicochemical Properties of Environmental Chemicals Using Molecular Fingerprints and Machine Learning. *J. Chem. Inf. Model.* **2017**, *57*, 36–49.

(50) He, H.; Garcia, E. Learning from Imbalanced Data. *IEEE TKDE* **2009**, *21*, 1263–1284.

(51) Cortes-Ciriano, I.; Bender, A. Improved Chemical Structure–Activity Modeling Through Data Augmentation. *J. Chem. Inf. Model.* **2015**, *55*, 2682–2692.

Graphical TOC Entry