Journal of
CHEMICAL INFORMATION
AND MODELING

Article

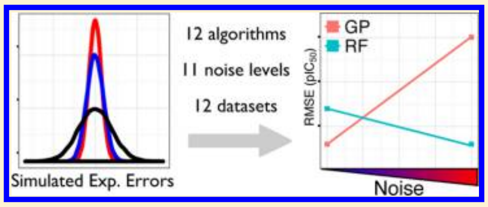
pubs.acs.org/jcim

Comparing the Influence of Simulated Experimental Errors on 12 Machine Learning Algorithms in Bioactivity Modeling Using 12 Diverse Data Sets

Isidro Cortes-Ciriano,*,[†] Andreas Bender,[‡] and Thérèse E. Malliavin[†]

[†]Département de Biologie Structurale et Chimie, Institut Pasteur, Unité de Bioinformatique Structurale, CNRS UMR 3825, 25, rue du Dr Roux, 75015 Paris, Ile de France, France

[‡]Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Cambridge CB2 1EW, United Kingdom



ABSTRACT: To date, no systematic study has assessed the effect of random experimental errors on the predictive power of QSAR models. To address this shortage, we have benchmarked the noise sensitivity of 12 learning algorithms on 12 data sets (15,840 models in total), namely the following: Support Vector Machines (SVM) with radial and polynomial (Poly) kernels, Gaussian Process (GP) with radial and polynomial kernels, Relevant Vector Machines (radial kernel), Random Forest (RF), Gradient Boosting Machines (GBM), Bagged Regression Trees, Partial Least Squares, and *k*-Nearest Neighbors. Model performance on the test set was used as a proxy to monitor the relative noise sensitivity of these algorithms as a function of the level of simulated noise added to the bioactivities from the training set. The noise was simulated by sampling from Gaussian distributions with increasingly larger variances, which ranged from zero to the range of pIC_{50} values comprised in a given data set. General trends were identified by designing a full-factorial experiment, which was analyzed with a normal linear model. Overall, GBM displayed low noise tolerance, although its performance was comparable to RF, SVM Radial, SVM Poly, GP Poly, and GP Radial at low noise levels. Of practical relevance, we show that the *bag fraction* parameter has a marked influence on the noise sensitivity of GBM, suggesting that low values (e.g., 0.1−0.2) for this parameter should be set when modeling noisy data. The remaining 11 algorithms display a comparable noise tolerance, as a smooth and linear degradation of model performance is observed with the level of noise. However, SVM Poly and GP Poly display significant noise sensitivity at high noise levels in some cases. Overall, these results provide a practical guide to make informed decisions about which algorithm and parameter values to use according to the noise level present in the data.

■ INTRODUCTION

Computational chemogenomic[1] techniques capitalize on bioactivity data to quantitatively predict and better understand unknown interactions between small molecules and their biomolecular targets. The development of these techniques has been mainly fostered by (i) the increase of computing resources and the availability of scalable machine learning software and (ii) the advent of high-throughput technologies, which have contributed to a vast increase of proprietary and public bioactivity data.[2,3] Although public bioactivity databases have grown in size steadily over the past decade, detailed information about the assays used and the experimental errors of the measurements are generally not reported. The question then arises how the experimental errors (or the lack thereof) should be included in the generation and validation of *in silico* predictive models and to which extent the quality of the data affects models predictive ability on new molecules. These issues need to be addressed prior to model training, e.g. which algorithms are robust to noisy input data and to which extent, and further downstream in the modeling pipeline, e.g. how should bioactivity models be validated according to the noise of the data. Here, we consider experimental error, or simply noise, as the random error of a measured variable, e.g. IC_{50} values.[4]

The quality of the data can be determined by the divergence of the average value of the experimental replicates (i.e., sample mean) with respect to the true bioactivity value, which would correspond to the average value of an infinite number of replicates (i.e., the population mean). The sample standard deviation decreases with the number of replicates (assuming that the experimental errors follow a Gaussian distribution). Thus, the difference between the population and the sample mean will decrease as the number of replicates increases, leading to a more precise estimation of the true bioactivity value. Therefore, in practice, the number of replicates and their standard deviation can serve to determine the quality of a data set. In this line, Wenlock et al.[5] have benchmarked the influence of the quality of the data in the generation of drug metabolism and pharmacokinetic models using data sets from AstraZeneca. The authors defined high-quality data as those bioactivity values measured in replicates with a standard deviation below a given threshold. This study showed that the quality of the training data is correlated to model performance on external molecules.

Whereas discarding those data points measured only once would constitute a reasonable cleaning step in the data

Received: February 23, 2015
Published: June 3, 2015

DOI: 10.1021/acs.jcim.5b00101

Table 1. Algorithms Benchmarked in This Study[a]

learning paradigm	algorithm	parameters and values used in CV	ref
Kernel	Gaussian Process Radial Kernel (GP Radial)	$\sigma \in \{2^{-10},2^{-4}..2^2,2^4\}$; σ_d^2 noise variance: 0.001	16
Kernel	Gaussian Process Polynomial Kernel (GP Poly)	scale $\in\{2^{-10},2^{-4}..2^2,2^4\}$; degree $\in(k)\,{}^{6}_{k\,=\,2}$; $\sigma^2{}_d$: 0.001	16
Kernel	Relevant Vector Machines Radial Kernel (RVM Radial)	$\sigma\in \{2^{-6},2^{-4}..2^2,2^4\}$	33
Kernel	Support Vector Machines Radial Kernel (SVM Radial)	$\sigma\in \{2^{-10},2^{-4}..2^2,2^4\}$; $C \in \{2^{-10},2^{-4}..2^2,2^4,10,100\}$	34
Kernel	Support Vector Machines Polynomial Kernel (SVM Poly)	scale $\in\{2^{-10},2^{-4}..2^2,2^4\}$; offset: 0; degree $\in(k)\,{}^{6}_{i\,=\,2}$; $C \in \{2^{-10},2^{-4}..2^2,2^4,10,100\}$	34
Ensemble: bagging	Bagged CART Regression Trees (Tree bag)		35
Ensemble: boosting	Gradient Boosting Machines (GBM)	learning rate (ν) $\in\{0.04, 0.08, 0.12, 0.16\}$; n_{trees}: 500; tree complexity (t_c): 25; bag fraction (η): 0.5	36,37
Ensemble: bagging	Random Forest (RF)	n_{trees}: 500	38
Linear	Partial Least Squares (PLS)		39
k-Nearest Neighbors (NN)	5-NN	$N_{neighbors}$: 5	40
k-Nearest Neighbors	10-NN	$N_{neighbors}$: 10	40
k-Nearest Neighbors	20-NN	$N_{neighbors}$: 20	40

[a] The third column indicates the parameters that were tunned using grid search and cross-validation (CV). The default values were used for those parameters not indicated therein.

Table 2. Data Sets Modelled in This Study[a]

data set label	biological end point	data points	activity range (pIC_{50})	bioactivity standard deviation (pIC_{50})	data source	ref
COX-1	human cyclooxygenase-1	1347	4.00−9.00	0.90	ChEMBL 16	9
COX-2	human cyclooxygenase-2	2312	4.00−10.70	1.20	ChEMBL 16	9
DHFR rat	rat dihydrofolate reductase	760	4.00−9.50	1.25	ChEMBL 16	41
DHFR homo	human dihydrofolate reductase	744	4.00−9.72	1.32	ChEMBL 16	41
F7	human factor-7	344	4.04−8.18	0.97	US20050043313	42
IL4	human interleukin-4	599	4.67−8.30	0.57	WO 2006/133426 A2	42
MMP2	human matrix metallopeptidase-2	443	5.10−10.30	0.99	WO2005042521	42
P61169	rat D_2 dopamine receptor	1968	4.00−10.22	1.02	ChEMBL 19	2
P41594	human metabotropic glutamate receptor 5	1381	4.00−9.40	1.14	ChEMBL 19	2
P20309	human muscarinic acetylcholine receptor M3	779	4.00−10.50	1.70	ChEMBL 19	2
P25929	human neuropeptide Y receptor type 1	501	4.11−10.70	1.40	ChEMBL 19	2
P49146	human neuropeptide Y receptor type 2	561	4.00−10.15	1.27	ChEMBL 19	2

[a] These data sets can be found in the references indicated in the last two columns.

collection phase,[5] in practice, this might lead to a marked decrease in the number of data points available for modeling, thus compromising the generation of statistically robust models. Therefore, it is paramount to control the trade-off between the quality and the size of the data. Two recent publications[6,7] have analyzed the variability of pK_i and pIC_{50} values from ChEMBL. The authors reported standard deviations for heterogeneous pIC_{50} and pK_i values of 0.68 and 0.54, respectively. In practice, these average experimental errors for public pIC_{50} and pK_i data, or the experimental errors of each data point when available,[8] can serve to assess whether the predictive power of the models is realistic or not.[8−11] In this line, Brown et al.[11] provided practical rules-of-thumb to evaluate the maximum R^2 (coefficient of determination) values attainable for the observed against the predicted pIC_{50} values for a set of compounds as a function of the range of pIC_{50} values considered and of the number of data points. This scheme was extended by Cortes-Ciriano et al.[10] by also considering the distribution of these pIC_{50} values and by proposing to calculate the distribution of minimum RMSE and maximum R^2 values given the quality of the data. These distributions of the maximum values for correlation metrics (e.g. R^2, R_0^2, or Q^2), and of the minimum values in the case of RMSE, can serve to assess whether the predictive power of the models is justified by the quality of the underlying training data, as well as to quantify the probability of obtaining a given R^2 or RMSE value. Thus, the distributions of maximum and minimum values can be regarded as sampling distributions for the validation metrics.

Although there exist algorithms to handle noisy input data,[12−16] the vast majority of the models reported in the medicinal chemistry literature are still based on algorithms that (i) treat the dependent variable as definite point values and (ii) that do not consider the experimental errors of the input data. Most machine learning algorithms have been developed assuming noise-free input data.[17] Thus, their application to real-world problems, where noisy data sets are prevalent, might lead to overfitting,[18] and thus to a decrease of model performance on external data. Assessing the magnitude of this decrease and the robustness to noise of different learning paradigms has been subject of intense investigation in the machine learning community.[17,19−25] Most of these studies have dealt with classification problems.[19,20,22,26] Nettleton et al.[26] compared the tolerance to noise, both on the descriptors and on the class labels, of the following classifiers on 13 highly unbalanced data sets: (i) Naive Bayes,[27] (ii) C4.5 decision trees,[28] (iii) IBk instance-based learner,[29] and (iv) Sequential Minimal Optimization (SMO) Support Vector Machines

DOI: 10.1021/acs.jcim.5b00101
J. Chem. Inf. Model. 2015, 55, 1413−1425

(SVM).[30] The authors found Naive-Bayes as the most robust algorithm, and SMO SVM the most sensitive, in agreement with Abhinav et al.[17] Interestingly, the authors showed that noise in the labels affects to a greater extent the performance of the learners when compared to noise in the descriptors. These results are reminiscent of the work by Norinder et al.[31] Therein, the authors benchmarked the tolerance to noisy chemical descriptors of decision tree ensembles across 16 QSAR data sets, finding that, in practice, the introduction of uncertainty in chemical descriptors does not reduce model performance.

To date, no systematic study has assessed the effect of random experimental errors of bioactivities on the predictive power of commonly used learning methods in QSAR. The present contribution aims at addressing this shortage. We recently compared the influence of the experimental errors on the predictive power of regression Gaussian Process (GP) models, finding that the radial kernel appears more robust to noisy input data than polynomial kernels,[10] which agrees with the machine learning literature.[32] Here, we extend this study by evaluating the influence of the experimental errors on the predictive power of 8 commonly used machine learning algorithms in a robust statistical manner (Table 1). The 8 machine learning algorithms, covering 5 learning paradigms, gave rise to 12 models as some parameters vary for a given method, e.g. kernel type. For the sake of clarity, these 12 models will be referred to as algorithms or models throughout the rest of the manuscript. The learning paradigms and algorithms are, respectively (Table 1), as follows: (i) kernel methods: GP (radial and polynomial kernel), SVM (radial and polynomial kernel), and Relevant Vector Machines (RVM) (radial kernel), (ii) ensemble bagging methods: Random Forest (RF) and Bagged CART Regression Trees (Tree bag), (iii) ensemble boosting methods: Gradient Boosting Machines (GBM), (iv) linear methods: Partial Least Squares (PLS), and (v) *k*-Nearest Neighbor (NN) learning (5-NN, 10-NN, and 20-NN). We used 12 QSAR data sets reporting compound potency as pIC_{50} values (Table 2). Chemical structures were encoded with Morgan fingerprints and 1-D and 2-D physicochemical descriptors. For each triplet (data set, algorithm, descriptor type) we trained each of the 12 models 11 times, each time with an increasingly higher level of simulated noise added to the pIC_{50} values from the training set. Model performance on the test set, quantified by the RMSE values for the observed against the predicted pIC_{50} values, was used as a proxy to assess the noise sensitivity of the 12 algorithms explored here. In order to identify general trends in a statistically sound manner and thus to assess the robustness of these algorithms with respect to the level of noise in the input data, we designed a balanced fixed-effect full-factorial experiment with replications. This experimental design was analyzed with a normal linear model using the RMSE values on the test set as the dependent variable.

■ METHODS

Data Sets. We gathered a total of 12 QSAR data sets from the literature (references given in Table 2) and from ChEMBL database version 19.[2] All data sets report compound potency as IC_{50} values. These values were modeled in a logarithmic scale ($pIC_{50} = -\log_{10} IC_{50}$). The size of the data sets range from 334 data points (Human Factor 7) to 2312 (Cyclooxygenase 2). Detailed information about these data sets can be found in Table 2.

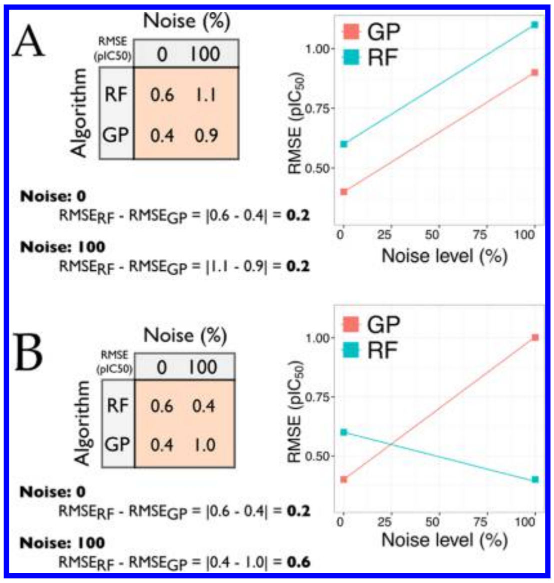


Figure 1. Illustration of two-way interactions between two-level factors, namely *Algorithm*: RF and GP and *Noise*: 0 and 100. There is interaction between two factors (**A**) when the difference of the mean RMSE values (response variable) across the levels of a factor (*e.g. Algorithm*) does not change across the levels of a second factor (*e.g. Noise*). In the example, the difference in performance between GP and RF is the same across all levels of factor *Noise*. This is illustrated by the presence of parallel lines. By contrast, the presence of nonparallel lines (**B**) indicates that the performance of GP and RF changes depending on the noise level. Thus, GP outperforms RF at noise level 0%, whereas RF outperforms GP at noise level 100%.

Molecular Representation. The function *StandardiseMolecules* from the R package *camb*[43] was used to normalize all chemical structures using the default options. This normalization step is crucial for the generation of compound descriptors, as the values of most of them (except for e.g. heavy atom counts) depend on a consistent representation of molecular properties such as aromaticity of ring systems, tautomer representation, or protonation states.

Compound Descriptors. Compounds were encoded with circular Morgan fingerprints[44,45] calculated using RDkit (release version 2013.03.02).[46,47] Morgan fingeprints encode compound structures by considering radial atom neighborhoods.[44] The choice of Morgan fingerprints as compound descriptors was motivated by the high retrieval rates obtained with these fingerprints in benchmarking studies of compound descriptors.[48,49] The size of the fingerprints was set to 256 bits, whereas the maximum radius of the substructures considered was set to 2 bonds. To calculate the fingerprints for a given compound, each substructure in that compound, with a maximal diameter of four bonds, was assigned to an unambiguous identifier. Subsequently, these substructures were mapped into a hashed array of counts. The position in the array where each substructure was mapped was given by the modulo of the division of the substructure identifier by the fingerprint size. A total of 188 1-D and 2-D physicochemical descriptors was computed with RDkit (release version 2013.03.02).[46]

Model Generation. The function *RemoveNearZeroVarianceFeatures* from the R package *camb* was used to remove those descriptors displaying constant values across all compounds (near-zero variance descriptors) using a cutoff value equal to 30/1.[43,50,51] Subsequently, the remaining descriptors were centered to zero mean and scaled to unit variance (z-scores) with the function *PreProcess* from the R package *camb*.

Grid-search with 5-fold cross-validation (CV) was used to optimize the model parameters.[18] The whole data set was split into 6 folds of the same size by stratified sampling of the pIC_{50} values. One fold, 1/6, was withheld as the test set and served to assess the predictive power of the models. The remaining folds, 5/6, constituted the training set and were used to optimize the values of the parameters in the following way. For each combination of parameter values in the grid, a model was trained on 4 folds from the training set, and the values for the remaining fold were then predicted. This procedure was repeated 5 times, each time holding out a different fold. The values of the parameters exhibiting the lowest average RMSE value across these 5 repetitions was considered as optimal. Subsequently, a model was trained on the whole training set, using the optimized values for the parameters. The predictive power of this model was assessed on the test set by calculating the RMSE value for the observed against the predicted bioactivities.

We run five replicates (models) for all factor level combinations. The training and test sets for each replicate were composed of different subsets of the complete data set. In order to make the results comparable on a given data set for a given replicate, all models were trained using the same fold composition. Thus, for a given data set and replicate the same test set was used to assess the predictive power of all algorithms at all noise levels. We note in particular that simulated noise was never added to the test sets.

Machine Learning Algorithms. Given a data set $D = \{\mathbf{X},\mathbf{y}\}$where $\mathbf{X} = \{\mathbf{x}^i\}_{i=1}^n$ is the set of compound descriptors and $\mathbf{y} = \{y^i\}_{i=1}^n$ is the vector of observed bioactivities, the aim of supervised learning is to find the function (model) underlying D, which can be then used to predict the bioactivity for new data points, $\mathbf{x}_{\mathbf{new}}$. In the following subsections we briefly summarize the theory behind the algorithms explored in this study.

Kernel Methods. Kernel functions, statistic covariances,[52] or simply kernels permit the computation of the dot product between two vectors, $\mathbf{x},\mathbf{x}' \in \mathbf{X}$, in a higher dimensional space, F (potentially of infinite dimension), without explicitly computing the coordinates of these vectors in F

$$\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = K(\mathbf{x}, \mathbf{x}') \tag{1}$$

where $\phi(\mathbf{x})$ is a mapping function from X to F, ϕ: $X \to F$. This is known as the kernel trick.[53] Thus, the value of the kernel K applied on the input vectors $\mathbf{x}$ and $\mathbf{x}'$ is equivalent to their dot product in F. In practice, this permits to apply linear methods based on dot products, e.g. SVM, in F while using $\mathbf{X}$ in the calculations (thus, not requiring to compute the coordinates of the input data in F). This is computationally less expensive than the explicit calculation of the coordinates of $\mathbf{X}$ in F, which in some cases might not even be possible. The linear relationships learned in F are generally nonlinear in the input space. Moreover, kernel methods are extremely versatile, as the same linear method, e.g. SVM, can learn diverse complex nonlinear functions in the input space because the functional form is controlled by the kernel, which in turn can be adapted to the data by the user.

The formulas for the kernels used in this study are

$$\text{Polynomial Kernel: } K(\mathbf{x}, \mathbf{x}') = (\text{scale } \mathbf{x}^{\mathbf{T}}\mathbf{x}' + \text{offset})^{degree} \tag{2}$$

$$\text{Radial Kernel: } K(\mathbf{x}, \mathbf{x}') = e^{-(\|\mathbf{x}-\mathbf{x}'\|^2/(2l^2))} \tag{3}$$

where $\|\mathbf{x}-\mathbf{x}'\|^2$ is the squared Euclidean distance, and $\mathbf{x}^{\mathrm{T}}$ is the transpose of $\mathbf{x}$. In the following, we will present different kernel methods used in the present work.

Gaussian Process (GP). In Bayesian inference,[54] the experimental data is used to update the *a priori* knowledge assumed for a certain problem. In the context of supervised learning, we *a priori* assume a distribution over the functions candidate to model the data, i.e. the *prior* distribution. The *prior* is then updated with the training examples, which yields the posterior probability distribution

$$P(GP(\mathbf{x})|D) \propto P(\mathbf{y}|GP(\mathbf{x}), \mathbf{X})P(GP(\mathbf{x})) \tag{4}$$

where (i) $P(GP(\mathbf{x})|D)$ is the *posterior* probability distribution giving the bioactivity predictions; (ii) the likelihood $P(\mathbf{y}|GP(\mathbf{x}), \mathbf{X})$ is the probability of the observations, $\mathbf{y}$, given the training set, $\mathbf{X}$, and the model $GP(\mathbf{x})$; and (iii) $P(GP(\mathbf{x}))$ is the *prior*.

GP[16] are a stochastic process that, similar to a multivariate Gaussian distribution, defined by its mean value and covariance matrix, is fully specified by its mean function, μ (usually the zero function), and its covariance function, $\mathbf{C}_X$

$$GP(\mathbf{x}) \sim \mathcal{N}(\mu, \mathbf{C}_X + \sigma_d^2\delta(\mathbf{x}_i, x_j))\ (i, j \in i ,..., n) \tag{5}$$

where $\delta(\mathbf{x}_j,x_k)$ is the Kronecker delta function, and σ_d^2 is the noise of the input data, which is assumed to be normally distributed with mean zero. $\mathbf{C}_X$ is obtained by applying a positive definite kernel function to $\mathbf{X}$, $\mathbf{C}_X = Cov(\mathbf{X})$. The function values for any set of input vectors follow a multidimensional normal distribution, and, therefore, the bioactivity value, y_{new}, for a new input vector, $\mathbf{x}_{\mathbf{new}}$, will also follow a Gaussian distribution defined by[54,55]

$$P(y_{new}) \sim N(\mu_{y_{new}} = \mathbf{k}^{\mathbf{T}}\mathbf{C}_{\mathbf{X}}^{-1}\mathbf{y}, \sigma_{y_{new}}^2 = m - \mathbf{k}^{\mathbf{T}}\mathbf{C}_{\mathbf{X}}^{-1}\mathbf{k}) \tag{6}$$

where $\mathbf{k} = Cov(\mathbf{X},\mathbf{x}_{\mathbf{new}})$. The best estimate for the bioactivity of $\mathbf{x}_{\mathbf{new}}$ is the average value of $\mathrm{P}(y_{new})$, namely $\mu_{y_{new}} = \langle P(y_{new})\rangle$.

As can be seen in eq 6, those input vectors in $\mathbf{X}$ similar to $\mathbf{x}_{\mathbf{new}}$, contribute more to the prediction of y_{new}, as $\mathbf{y}$ is weighted by $\mathbf{k}^{\mathbf{T}}$. The predicted variance, $\sigma_{y_{new}}^2$, corresponds to the difference between the *a priori* knowledge about $\mathbf{x}_{\mathbf{new}}$: $m = Cov(\mathbf{x}_{\mathbf{new}}, \mathbf{x}_{\mathbf{new}})$ and what can be inferred about $\mathbf{x}_{\mathbf{new}}$ from similar input vectors: $\mathbf{k}^{\mathbf{T}}\mathbf{C}_{\mathbf{X}}^{-1}\mathbf{k}$.

Support Vector Machines (SVM). SVM[34,56] fit a linear model in a higher dimensional dot product feature space, F, of the form

$$f(\mathbf{x}|\mathbf{w}) = \langle \mathbf{w}^T\phi(\mathbf{x})\rangle + \alpha_0 \tag{7}$$

where $\mathbf{w}$ is a vector of weights $\in F$. The kernel trick can be applied if $\mathbf{w}$ can be expressed as a linear combination of the training examples, namely $\mathbf{w} = \sum_{i=1}^n \alpha_i\phi(\mathbf{x}_i)$. Given the definition of kernel given above, eq 7 can be rewritten as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle + \alpha_0 = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + \alpha_0 \tag{8}$$

The optimization of the α_i values is usually performed by applying Lagrangian multipliers (dual formulation)

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

subject to $\sum_{i=1}^{n}\alpha_i y_i = 0$ and $0 \le \alpha_i \le C$. C is a regularization parameter that penalizes for incorrect predictions during training. Thus, the larger the value of C, the larger this penalty.

Relevance Vector Machines (RVM). RVM[33] follow a similar formulation to SVM with the exception that the weights are inferred from the data in a Bayesian framework by defining an explicity prior probability distribution, normally Gaussian, on the parameters α_i:

$$f(\mathbf{x}|\alpha) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + \alpha_0 \tag{10}$$

This formulation leads to sparse models as a large fraction of the weights are sharply distributed around zero. Thus, only a small fraction of examples from $\mathbf{X}$ (the Relevance Vectors) are used when making predictions using eq 10.

Ensemble Methods. Ensemble methods use multiple weak simple models (base learners) to obtain a meta-model with higher predictive power than it would be possible to obtain with any of the models used individually. Thus, building a model ensemble consists of (i) training individual models on (subsets) of the training examples and (ii) integrating them to generate a combined prediction. Although it is possible to build model ensembles using different machine learning algorithms (i.e., heteroensembles) as base learners, e.g. model stacking,[9,57] homoensembles such as decision tree-based ensembles are predominant in the literature. The following subsection briefly presents the ensemble methods used in this study.

Bagging: Bagged CART Regression Trees (Tree Bag). Bootstrap aggregating or Bagging is a technique that averages the prediction of a set of high-variance base learners (normally regression trees, e.g. CART),[35] each trained on a bootstrap sample, b, drawn with replacement from the training data. Thus, bagging leads to higher stability and predictive power with respect to the individual base learners and reduces overfitting.[57] In practice, high-variance and low-bias algorithms, such as regression trees, have proved to be very well suited for bagging.[57] The model can be formulated as

$$f(\mathbf{x}) = \frac{1}{B}\sum_{b=i}^{B} T_b(\mathbf{x}) \tag{11}$$

where $T_b(\mathbf{x})$ corresponds to the tree base learner trained on the bth bootstrap sample.

Boosting: Gradient Boosting Machines (GBM). Boosting[37,57,58] differs from bagging in that the base learners, here regression trees, are trained and combined sequentially. At each iteration, a new base-learner is trained on the bth bootstrap sample, G_b, and added to the ensemble trying to minimize the loss function associated with the whole ensemble. The loss function, Ψ, can be e.g. the squared error loss, i.e. the average of the square of the training residuals: $\Psi(y,f) = ((1)/(n))\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$.

The final model is given by

$$f(\mathbf{x}) = \sum_{b=i}^{B} w_b G_b(\mathbf{x}) \tag{12}$$

where $G_b(\mathbf{x})$ is the base learner trained on the bth bootstrap sample, w_b is its weight, and B is the total number of iterations and trees. The weight for a base learner, w_b, is, thus, proportional to its prediction accuracy. The update rule for the model can be written as

$$f_b(\mathbf{x}) = f_{b-1}(\mathbf{x}) + \nu w_b G_b(\mathbf{x});\ 0 < \nu \le 1 \tag{13}$$

where $f_b(\mathbf{x})$ corresponds to the ensemble at iteration b, and ν corresponds to the learning rate (see below). Deepest gradient descent is applied at each iteration to optimize the weight w_b for the new base learner as follows:

$$w_b = \min_{w} \sum_{i=1}^{n} \Psi(y_i, f_{b-1}(\mathbf{x}) + G_b(\mathbf{x})) \tag{14}$$

To minimize the risk of overfitting of GBM, several procedures have been proposed. The first one consists of training the individual base learners on bootstrap samples of smaller size than the training set. The relative size of these samples with respect to that of the training set is controlled by the parameter *bag fraction* or η. A second procedure is to reduce the impact of the last tree added to the ensemble on the minimization of the loss function by adding a regularization parameter, shrinkage or learning rate (ν). The underlying rationale is that sequential improvement by small steps is better than improving the performance substantially in a single step. Likewise, the effect of an inaccurate learner on the whole ensemble is thus reduced. Another way to reduce overfitting is to control the complexity of the trees by setting the maximum number of nodes of the base learners with the parameter tree complexity (t_c). Finally, the number of iterations, i.e. number of trees in the ensemble (n_{trees}), also needs to be controlled. The training error decreases with the number of trees, although a high number of iterations might lead to overfitting.[37]

Random Forest (RF). Like in bagging, RF[38] build an ensemble (forest) of regression tress and average their predictions

$$f(\mathbf{x}) = \frac{1}{B}\sum_{b=i}^{B} TF_b(\mathbf{x}) \tag{15}$$

where $TF_b(\mathbf{x})$ corresponds to the tree base learner in the forest trained on the bth bootstrap sample. The difference with bagging is that the node splitting is performed using only a subset of descriptors randomly chosen. This additional level of randomization decorrelates the trees in the forest leading, in practice, to a predictive power comparable to boosting.[57]

In QSAR, RF have been shown to be robust with respect to the parameter values. In practice, a suitable choice of the number of trees (n_{trees}) was shown to be 100, as higher values do not generally lead to significantly higher predictive power.[59,60]

Partial Least Squares Regression (PLS). Partial least-squares or *projection to latent structures*[39] is a multivariate modeling technique capable of extracting quantitative relationships from data sets where the number of descriptors, P, is much larger than the number of training examples, N. Multiple linear regression fails to model this type of data sets since for small (N/P) ratios, $\mathbf{X}$ is not a full rank matrix and its columns will be probably collinear (the "small N large P problem").[61] In Principal Components Regression (PCR), the principal components of $\mathbf{X}$ are taken as predictors, thus reducing P (dimensionality reduction) and the problem of multicollinearity. PLS extends this idea by simultaneously projecting both $\mathbf{X}$ and $\mathbf{y}$ to latent variables, with the constraint of maximizing the covariance of the projections of $\mathbf{X}$ and $\mathbf{y}$. Subsequently, the response variable is obtained on the latent vectors obtained on $\mathbf{X}$. We refer the reader to Abdi and Williams[62] for further details on PLS.

k-Nearest Neighbors (k-NN). The k-NN algorithm averages the response value over the k closest neighbors to estimate the response value for a data point as

$$f(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{16}$$

The Euclidean distance was used to find the k closest neighbors.

Machine Learning Implementation. Machine learning models were built in R using the wrapper packages *caret*[50] and *camb*.[43] The following R packages were used to train the machine learning algorithms considered here: (i) *kernlab*[63] for Support Vector Machines (SVM),[56] Relevance Vector Machines (RVM),[33] and Gaussian Processes (GP),[16] (ii) *gbm*[64] for Gradient Boosting Machines (GBM),[36] (iii) *class*[65] for k-Nearest Neighbors (KNN),[40] (iv) *pls*[66] for Partial Least Squares (PLS),[39] (v) *randomForest*[67] for Random Forest (RF),[38] and (vi) *ipred*[68] for bagged Classification And Regression Trees (CART).[35]

Simulation of Noisy Bioactivities. To assess the effect of random experimental errors on the predictive power of QSAR, 11 models *per* triplet (data set, algorithm, descriptor type) were trained, each of them with an increasingly larger level of noise, ϵ, added to the pIC_{50} values from the training set. Noise levels were simulated by sampling from a Gaussian distribution with zero mean and corresponding larger variance, σ^2_{noise}. The value of the variance across the 11 noise levels was defined as a function of the range of bioactivities considered in each data set

$$\{\sigma^2_{noise\ i}\}_{i=0}^{10} = (\max(pIC_{50}) - \min(pIC_{50}))*Noise_{level\ i} \tag{17}$$

where i is the index of the noise level, and $(\max(pIC_{50}) - \min(pIC_{50}))$ corresponds to the range of pIC_{50} values comprised in a given data set. $Noise_{level}$ was defined as

$$Noise_{level} = \{i/10\}_{i=0}^{10} \tag{18}$$

The first noise level corresponds to a variance of 0, i.e. no noise was added and, therefore, the bioactivity values corresponded to the reported pIC_{50} values. The bioactivity values for the training set, $\mathbf{Y}_{\mathbf{tr\ Noise}}$, were calculated as

$$\mathbf{Y}_{\mathbf{tr\ Noise\ i}} = \mathbf{Y}_{\mathbf{tr}} + \epsilon \sim N(0, \sigma^2_{noise\ i}) \tag{19}$$

where $\mathbf{Y}_{\mathbf{tr\ Noise\ i}}$ corresponds to the noisy pIC_{50} values for noise level i, $\mathbf{Y}_{\mathbf{tr}}$ corresponds to the reported pIC_{50} values, and ϵ corresponds to the vector containing the simulated noise. Therefore, $\mathbf{Y}_{\mathbf{tr\ Noise\ i}}$ was used as the dependent variable during model training.

The simulated noise is thus sampled from a Gaussian distribution with variance: $\sigma^2_{noise\ i}$. This choice makes the results comparable across data sets irrespective of the range of pIC_{50} values comprised in each of them. For convenience, noise levels will be reported in the following as percentage points (e.g., noise level 0.5 would correspond to 50%). We note in particular that this noise simulation method is sensitive to outliers (i.e., highly active or inactive compounds), as the range of pIC_{50} values would be considerably enlarged in their presence. Thus, we advise to remove outliers for the generation of the range of noise levels.

We are aware that the reported pIC_{50} values, which would correspond to noise level 0%, already contain random experimental errors. We are not overly concerned about this issue given that all models trained on a given data set need to deal with that base level of noise.

Experimental Design. In order to investigate the relative noise sensitivity of the 12 algorithms, a balanced fixed-effect full-factorial experiment with replications was designed.[69] The following four factors were considered, namely: data set (*Data set*), noise level (*Noise*), descriptor type (*Descriptor type*), and learning algorithm (*Algorithm*). Given that the factor *Data set* has an influence on model performance, as some data sets are better modeled than others, but it is irrelevant to the effect of interest (i.e., the effect of noise across learning algorithms irrespective of the data set), it was considered as a blocking factor.[69] Similarly, the factor *Descriptor type* was also added as a blocking factor.

This factorial design was studied with the following linear model

$$\begin{aligned} RMSE_{i,j,k,l,m\ test} = {} & Data\ set_i + Descriptor\ type_j + Noise_k \\ & + Algorithm_l + (Noise{*}Algorithm)_{kl} + \mu_0 \\ & + \epsilon_{i,j,k,l,m} \end{aligned} \tag{20}$$

$$(i \in \{1, ..., N_{data\ sets} = 12\}; \quad j \in \{1, ..., N_{Descriptor\ type} = 2\}; \quad k \in \{1, ..., N_{noise\ levels} = 11\}; \quad l \in \{1, ..., N_{algorithms} = 12\}; \quad m \in \{1, ..., N_{resamples} = 100\})$$

where the response variable, $RMSE_{i,j,k,l,m\ test}$, corresponds to the RMSE values on the test set. $Data\ set_i$, $Descriptor\ type_j$, $Noise_k$, and $Algorithm_l$ are the main effects, and $Noise{*}Algorithm$ corresponds to the interaction term between the learning algorithm and the noise level.The factor levels Random Forest (*Algorithm*), F7 (*Data set*), Morgan fingerpints (*Descriptor type*), and Noise: 0 (*Noise level*) were used as reference factor levels to calculate the intercept term of the linear model, μ_0, which is simply the mean $RMSE_{test}$ value for this combination of factor levels. The coefficients (slopes) for the other factor level combinations, e.g. GP:Noise 20%, correspond to the difference between their mean $RMSE_{test}$ value and the intercept. The error term, $\epsilon_{i,j,k,l,m}$, corresponds to the random error of each $RMSE_{test}$ value, which are defined as $\epsilon_{i,j,k,l,m} = RMSE_{i,j,k,l,m\ test} - \overline{RMSE_{i,j,k,l}}$. These errors are assumed to (i) be mutually independent, (ii) have zero expectation, and (iii) have constant variance.

One model was trained for all factor level combinations, giving rise to 15,840 models (12 learning algorithms × 11 noise levels × 12 data sets × 2 descriptor types × 5 replications). The predictive power of the models, which serves as a proxy to evaluate the noise sensitivity of the algorithms, was assessed on the test set and quantified by the RMSE value, $RMSE_{i,j,k,l,m=1\ test}$, for the observed against the predicted bioactivities. Bootstrapping[70] was used to generate 100 resamples ($N_{replications}$) for these $RMSE_{i,j,k,l,m\ test}$ values (*i.e.* $RMSE_{i,j,k,l,m=2:100\ test}$), thus ensuring a balanced experimental design. Therefore, the total number of observations considered in the linear model was 1,584,000 (15,840 trained models × 100 resamples each). The significance level was set to 5%. The normality and homoscedasticity assumptions of the linear model were respectively assessed with (i) quantile-quantile (Q-Q) plots and (ii) by visual inspection of the $RMSE_{test}$ distributions and by plotting the $RMSE_{test}$ values against the residuals.[69]

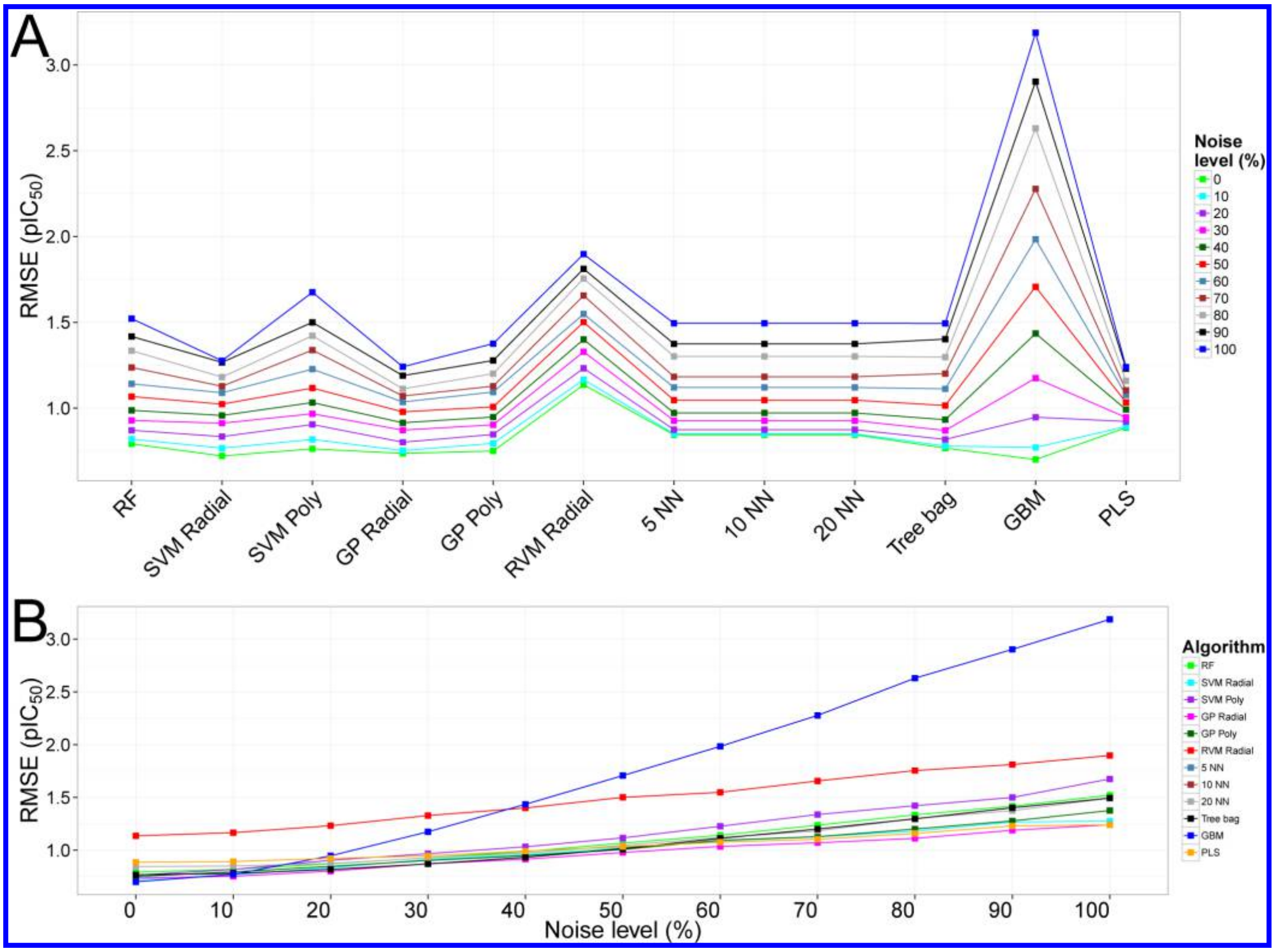


Figure 2. A,B. Interaction plots. Mean $RMSE_{test}$ values across all data sets for two-way combinations of factors. The data set-specific intercept was subtracted from the $RMSE_{test}$ values in order to make the results comparable across the 12 data sets. Abbreviations. GP: Gaussian Process; GBM: Gradient Boosting Machine; *k*-NN: *k*-Nearest Neighbors; PLS: Partial Least Squares; Poly: Polynomial kernel; RF: Random Forest; Radial: Radial kernel; RMSE: root-mean-square error in prediction; RVM radial: Relevant Vector Machines with radial kernel; SVM: Support Vector Machines; Tree bag: Bagged CART Regression Trees.

The interaction term was introduced to assess the interaction effects between the factors *Algorithm* and *Noise*. Figure 1 illustrates the concept of two-way interactions (i.e., between two factors) with a toy example where both the factor *Algorithm* and the factor *Noise* have only two levels, namely *Algorithm*: *RF* and *GP* and *Noise*: 0 and 100. There is no interaction between two factors when the difference of the mean values of the dependent variable (in this case RMSE) across the levels of a factor (*e.g. Algorithm*) does not change across the levels of a second factor (*e.g. Noise*). In the example, this means that the difference in RMSE between RF and GP is the same irrespective of the level of noise (Figure 1A). Therefore, it could be concluded that the difference in performance between RF and GP is not affected by the level of noise. This can be easily shown in an interaction plot (Figure 1A right panel) by plotting the levels of the factor *Noise* against the RMSE values for GP and RF. In the absence of interaction, the lines connecting the points corresponding to the RMSE values for each algorithm along the levels of the factor *Noise* are parallel.

By contrast, in the presence of interaction (Figure 1B), the difference in RMSE between RF and GP would vary across the levels of the factor *Noise*. Therefore, the performance of the algorithms would depend on the level of noise, and the lines in the interaction plot would not be parallel (Figure 1B right panel). Consequently, it would not be possible to conclude about the effect of a single independent variable (e.g., factor *Algorithm*) on the RMSE (termed *main effects*), as the RMSE values would depend on the level of other factors (e.g. *Noise*). For instance, in Figure 1B, it would not be possible to state that RF perform better than GP because the RMSE values corresponding to each algorithm depend on the level of noise. Therefore, in the presence of interaction, the influence of a factor on the dependent variable has to be analyzed for each level of the second factor, which is known as analysis of *simple effects*. In the example (Figure 1B), this would correspond to stating that GP perform better than RF when no noise is present in the input data, whereas RF perform better than GP when the level of noise equals 100%.

■ RESULTS

The fitted linear model displayed an adjusted R^2 value (adjusted by the complexity - number of parameters - of the model) of 0.71 and a standard error for the residuals of 0.27. This analysis showed significant interaction between the factors *Noise* and

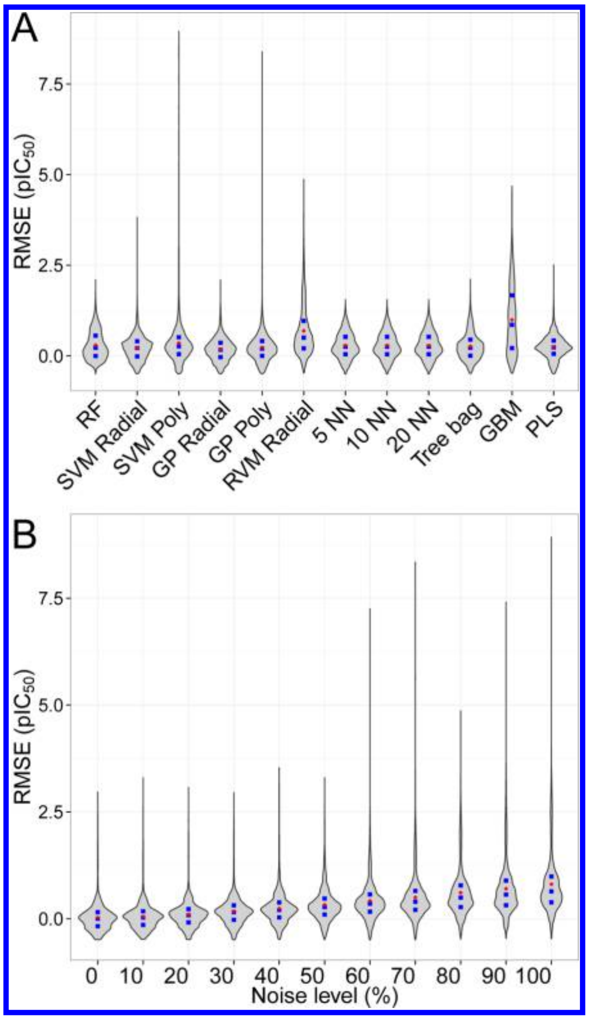


Figure 3. Violin plots. **A.** $RMSE_{test}$ values across all data sets, descriptor types, and noise levels for the 12 algorithms. **B.** $RMSE_{test}$ values across all data sets, descriptor types, and models for the 11 noise levels studied. The data set-specific intercept was subtracted from the $RMSE_{test}$ values in order to make the results comparable across the 12 data sets. Blue points indicate the median and the interquartile range (25th−75th percentile), whereas red points indicate the mean $RMSE_{test}$ value. Abbreviations. GP: Gaussian Process; GBM: Gradient Boosting Machine; *k*-NN: *k*-Nearest Neighbors; PLS: Partial Least Squares; Poly: Polynomial kernel; RF: Random Forest; Radial: Radial kernel; RMSE: root-mean-square error in prediction; RVM radial: Relevant Vector Machines with radial kernel; SVM: Support Vector Machines; Tree bag: Bagged CART Regression Trees.

Algorithm (*P*-value <0.001), thus indicating that the effect of *Noise* on the $RMSE_{test}$ values is not constant across the algorithms studied and *vice versa*. This is illustrated by the presence of nonparallel lines in the interaction plots (Figure 2A,B). Figure 3A shows the distributions of $RMSE_{test}$ values for the 12 learning algorithms across all replications, data sets, descriptor types, and noise levels, whereas Figure 3B reports the distributions of $RMSE_{test}$ values for the 11 noise levels considered across all data sets, descriptor types, algorithms, and replications. The values for the coefficients, namely slopes and intercept, and their *P*-values are reported in Table 3.

RVM Radial constantly displays the worst predictive power (Figure 2A), followed by GBM, Tree bag, 5-NN, 10-NN, and 20-NN. This effect can also be inferred from the high value for the slope corresponding to the factor level *RVM Radial*, namely 0.34 pIC_{50} units (Table 3, first column). Although low, the predictive power of RVM exhibits a smooth degradation with the level of noise comparable to the other methods, with the exception of GBM. This indicates that RVM are less sensitive to noise than GBM. Interestingly, GBM displays mean $RMSE_{test}$ values comparable to those obtained with RF, SVM Radial, GP Radial, GP Poly, and SVM Poly for noise levels 0 and 10% (Figure 2A), which is also indicated by the slope value for GBM, namely −0.09 (Table 3, first column). However, the mean $RMSE_{test}$ values significantly increase from noise level 20% upward, thus indicating that the performance of GBM highly depends on the noise level. The sensitivity to noise of the remaining 11 algorithms displays a smooth and linear degradation with the level of noise (Figure 2B), showing that these algorithms exhibit a comparable tolerance to noise and, thus, are less prone to overfitting than GBM.

The sensitivity to noise of boosting algorithms has been previously reported.[71] Introducing randomness in ensemble modeling by subsampling has proved efficient to increase the generalization ability of a model and to reduce its susceptibility to overfitting.[37,38,72] In GBM, randomness during training is introduced by controlling the fraction (*bag fraction*) of the training data randomly sampled without replacement that will be used to fit the next base tree learner at each consecutive learning iteration. The value for *bag fraction* is set to 0.5 by default.[37] To further understand the effect of this parameter on the noise tolerance of GBM in QSAR, we trained a model using Morgan fingerprints as compound descriptors for all *Algorithm-Noise* combinations across a wide range of *bag fraction* values (*bag fraction* $\in (k/10)_{k=1}^{10}$) for 7 data sets of diverse size, thus giving rise to 840 models (7 data sets * 12 algorithms * 10 *bag fraction* values). Figure 4 reports the mean $RMSE_{test}$ values for these models. Up to a noise level of 30%, the performance of all models is comparable across the range of *bag fraction* values explored. By contrast, from 30% onward the difference increases abruptly. In all cases, the mean $RMSE_{test}$ difference between models trained with *bag fraction* values of 1 and 0.1−0.2 increases with the noise level, reaching a difference value of ~1.5 pIC_{50} units at noise level 100%. Taken together, these data evidence that a proper tuning of the *bag fraction* parameter is required to palliate the noise sensitivity of GBM.

SVM Poly displayed low noise sensitivity at low noise levels, although the tolerance to noise conspicuously decreased at noise levels higher than 50%. This is illustrated by the interpoint distance in Figure 2A. This phenomenon was less marked for GP Poly and was not observed for GP Radial nor SVM Radial. Overall, the noise sensitivity of GP Poly is comparable to that of SVM Radial, GP Radial, and RF. Nevertheless, GP Poly and SVM Poly exhibit high noise sensitivity in some cases, as indicated by the corresponding tails in the violin plots in Figure 3A.

Similar to GBM, the machine learning community has reported the propensity of *k*-Nearest Neighbors to overfitting when modeling noisy data in classification settings.[73−76] The performance of 5-NN, 10-NN, and 20-NN was found comparable to that of Tree bag and PLS and lower than that of RF, SVM Radial, SVM Poly, GP Poly, and GP Radial. The sensitivity to noise of Tree bag, 5-NN, 10-NN, and 20-NN decreased more sharply at high noise levels, which can be observed by the interpoint distance in Figure 2A. As a rule of thumb, it is considered that the sensitivity to noise decreases with the increase of the number of neighbors (*k*).[77] Here, we did not observe this trend, as the noise sensitivity of 5-NN, 10-

Table 3. Values for the Slopes (Coefficients) and *P*-Values for the Fitted Linear Model[a]

factor level	slope	*P*-value	factor level	slope	*P*-value	factor level	slope	*P*-value
RF with Noise 0	0.60	<0.01	20 NN	0.05	<0.01	Noise 50	0.28	<0.01
SVM Radial	−0.07	<0.01	Tree bag	−0.02	<0.01	Noise 60	0.35	<0.01
SVM Poly	−0.03	<0.01	GBM	−0.09	<0.01	Noise 70	0.45	<0.01
GP Radial	−0.06	<0.01	PLS	0.09	<0.01	Noise 80	0.54	<0.01
GP Poly	−0.04	<0.01	Noise 10	0.03	<0.01	Noise 90	0.63	<0.01
RVM Radial	0.34	<0.01	Noise 20	0.08	<0.01	Noise 100	0.73	<0.01
5 NN	0.05	<0.01	Noise 30	0.14	<0.01	physicochemical descs.	−0.03	<0.01
10 NN	0.05	<0.01	Noise 40	0.20	<0.01			

factor level	slope	*P*-value	factor level	slope	*P*-value	factor level	slope	*P*-value
SVM Radial: Noise 10	0.02	<0.01	GP Radial: Noise 20	−0.01	<0.01	RVM Radial: Noise 30	0.06	<0.01
SVM Poly: Noise 10	0.03	<0.01	GP Poly: Noise 20	0.02	<0.01	5 NN: Noise 30	−0.05	<0.01
GP Radial: Noise 10	−0.01	0.03	RVM Radial: Noise 20	0.02	<0.01	10 NN: Noise 30	−0.05	<0.01
GP Poly: Noise 10	0.02	<0.01	5 NN: Noise 20	−0.05	<0.01	20 NN: Noise 30	−0.05	<0.01
RVM Radial: Noise 10	0.00	0.58	10 NN: Noise 20	−0.05	<0.01	Tree bag: Noise 30	−0.03	0.01
5 NN: Noise 10	−0.02	<0.01	20 NN: Noise 20	−0.05	<0.01	GBM: Noise 30	0.34	<0.01
10 NN: Noise 10	−0.02	<0.01	Tree bag: Noise 20	−0.03	<0.01	PLS: Noise 30	−0.08	<0.01
20 NN: Noise 10	−0.02	<0.01	GBM: Noise 20	0.17	<0.01	SVM Radial: Noise 40	0.04	<0.01
Tree bag: Noise 10	−0.02	<0.01	PLS: Noise 20	−0.04	<0.01	SVM Poly: Noise 40	0.07	<0.01
GBM: Noise 10	0.04	<0.01	SVM Radial: Noise 30	0.05	<0.01	GP Radial: Noise 40	−0.02	<0.01
PLS: Noise 10	−0.02	<0.01	SVM Poly: Noise 30	0.07	<0.01	GP Poly: Noise 40	0.00	0.54
SVM Radial: Noise 20	0.03	<0.01	GP Radial: Noise 30	0.00	0.89			
SVM Poly:	0.06	<0.01	GP Poly:	0.02	<0.01			

NN, and 20-NN remains constant across all noise levels, as illustrated by the parallel lines observed in Figure 2A. Therefore, *k*-NN appears robust to noise in the context of QSAR across the data sets studied. The performance of *k*-NN models was slightly lower than that of Tree bag across all noise levels (Figure 2A), whereas RF constantly displayed comparable predictive power (Figure 2A,B) to Tree bag. It is known that bagging reduces the variance of the final model.[57] In the case of RF, the additional layer of randomization is expected to decrease the sensitivity to noise and, thus, lead to

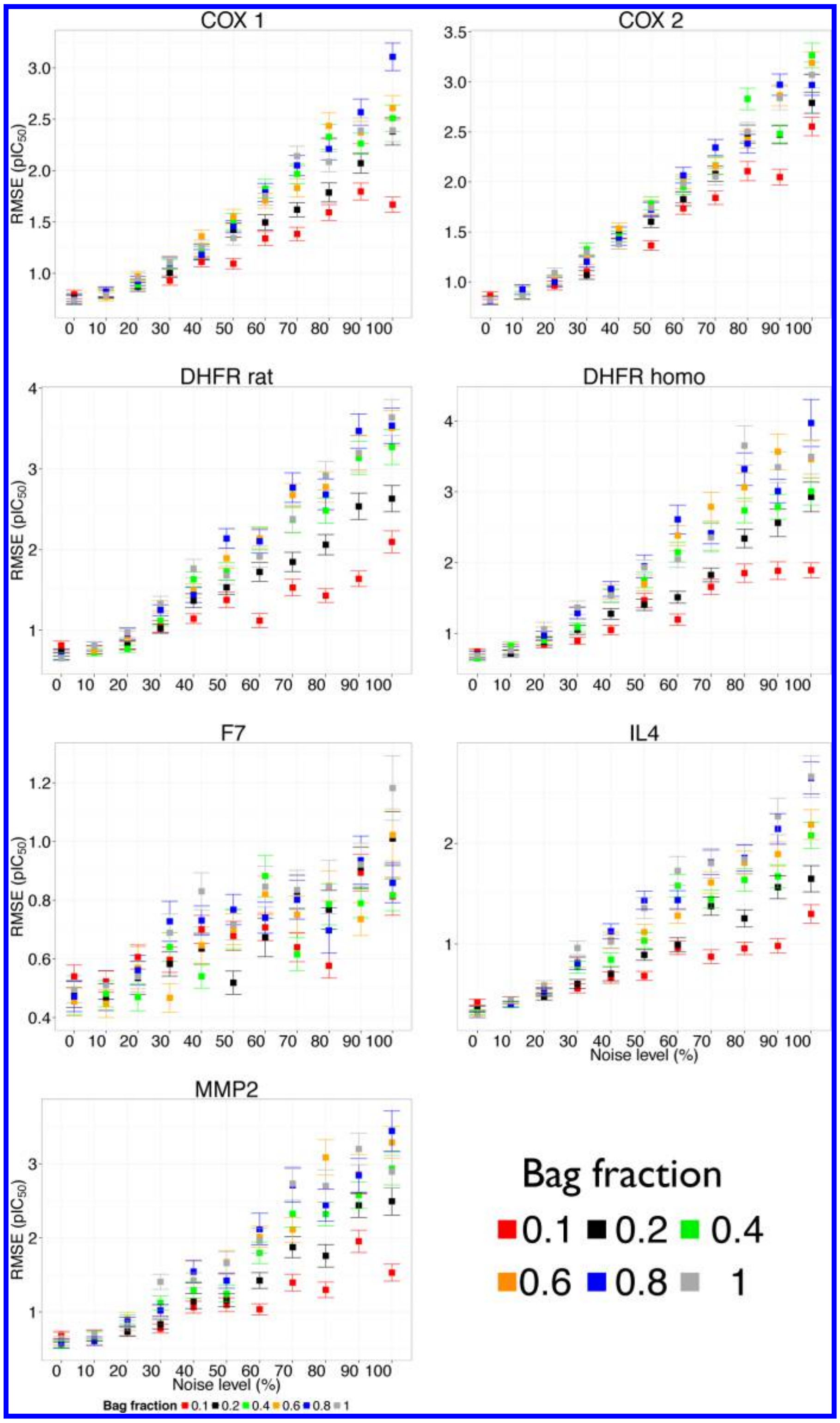


Figure 4. $RMSE_{test}$ values (mean ± std) for GBM models trained with increasingly higher *bag fraction* values across all *Noise−Algorithm−data set* combinations using Morgan fingerprints as compound descriptors. For low noise levels (up to 20%) the performance of all models is comparable irrespective of the *bag fraction* value. However, from noise level 30% upward, the mean $RMSE_{test}$ difference between models trained with bag fraction values of 1 and 0.1−0.2 increases with the noise level. Overall, these data suggest that the noise sensitivity of GBM highly depends on the *bag fraction* value.

higher predictive power on the test set. However, this effect was not observed across the algorithms, noise levels, data sets, and descriptors explored in this study.

Worth mentioning is the fact that at the highest noise levels explored, PLS displays the highest predictive power, as well as the smoothest degradation in performance as the level of simulated noise increases. This can be observed by the low interpoint distance in Figure 2A and by the low slope of the line corresponding to PLS (orange) in Figure 2B. Therefore, PLS appears more robust to noise than more algorithmically complex techniques such as RF.

DISCUSSION

We have benchmarked the noise sensitivitiy of 12 learning algorithms commonly used in QSAR, comprising 5 learning paradigms, on 12 data sets using two descriptor types, namely Morgan fingerprints (topological descriptors) and physicochemical-property-based descriptors. Model performance on the test set was used as a proxy to monitor the relative noise sensitivity of these algorithms as a function of the level of noise added to the bioactivities from the training set. The noise was simulated by sampling from Gaussian distributions with increasingly larger variances, which ranged from zero to the range of pIC_{50} values comprised in a given data set. Although the exploration of machine learning algorithms (and data sets) reported is not exhaustive, we have conducted robust statistical analyses, which have evidenced general trends about the behavior of these algorithms across different noise levels and, in the case of kernel methods, across kernel types. Overall, GBM displayed low tolerance to noisy bioactivities although its performance was comparable to RF, SVM Radial, SVM Poly, GP Poly, and GP Radial for low noise levels.

We note in particular that at noise level 0%, the lowest predictive power, excluding RVM Radial, is displayed by 5-NN, 10-NN, 20-NN, and PLS, which are the least algorithmically complex methods among the learning strategies explored. This also indicates that the relationship between the pIC_{50} values and the molecular properties presents a certain degree of nonlinearity, thus making the data sets used here suitable to benchmark the noise sensitivity of nonlinear algorithms. Similarly, it is important to note that the aim here is to assess the noise sensitivity of a set of algorithms covering diverse learning paradigms but not to compare their relative performance on these data sets (although all models displayed sufficient predictive power to be considered as statistically robust).[78]

In a previous publication,[10] it was reported the differential tolerance to noise of Gaussian Process models depending on the kernel chosen. Here, we found that both GP models with radial (GP Radial) and polynomial (GP Poly) kernels displayed high predictive power on the test set for low noise levels (0 and 10%). By contrast, the $RMSE_{test}$ values slightly increased in the case of GP Poly with the level of noise, which agrees with the machine learning literature.[10,17,32,57] This effect was more evident in the case of SVM models. In practice, it is advisible to use a low degree for the polynomial kernels when used with SVM, as polynomial kernels of higher degree are prone to overfitting and are less robust to noise.[57] From a practical standpoint, we observed that the noise tolerance of SVM Poly and GP Poly could be improved (data not shown) if the grid used to optimize the model parameters in cross-validation covers a wide range of values (Table 1). Therefore, we advocate to perform grid search across a broad range of parameter values for GP Poly(*scale*) and SVM Poly(*scale* and *C*).

Interestingly, the noise sentivity of SVM Radial and GP Radial was comparable, thus indicating that the variability in noise sensitivity for SVM and GP is more dependent on the kernel choice than on the machine learning technique. Although GP and SVM display comparable predictive power,[10] the Bayesian formulation of GP enables the inclusion of the experimental error of each data point as input to the model. This property of GP might be useful when modeling data sets in which the experimental errors for individual data points are reported.

Another interesting observation is the low noise tolerance of GBM under the default parameter settings, i.e. the bag fraction $\eta = 0.5$. These results are in line with Dietterich,[72] who reported that boosting displays higher performance than bagging in classification on noise-free data, whereas in the presence of noise bagging outperforms boosting. Therefore, these data indicate that careful attention should be given to the choice of parameter values when using GBM in QSAR.

It is important to note that noise in QSAR does not always correspond to random experimental errors. For instance, the purity of compounds can degrade over time, and their solubility in the assay medium is not always verified. Similarly, IC_{50} values depend on the assay conditions. Therefore, it is advisible to consider these sources of error prior to building QSAR models whenever possible. For a detailed review of the sources of errors in bioactivity data see ref 79. Concerning the quantification of the level of noise (quality) of bioactivity data, Wenlock et al.[5] considered as high quality data those data points whose replicate standard deviation was below an arbitrary cutoff value. Whereas information about the quality of the data might be available when using in-house data obtained in normalized experimental conditions, this is not generally possible for academic laboratories, as public databases lack detailed information about the assay protocols and the variation across experimental replicates. Therefore, assessing the quality of public data might not always be possible. In these cases, it is reasonable to consider respective standard deviations of 0.68 and 0.54 for heterogeneous pIC_{50} and pK_i values.[6,7]

Overall, this study provides a practical guide to make informed decisions about which algorithm and parameter values to use according to the noise level present in the data. As we have shown here, an inappropriate algorithmic (or kernel) choice can have a significant impact on the predictions on external molecules when modeling low quality data, even for low noise levels. Therefore, the quality of the data, be it estimated from replicates or from the literature,[6,7] should become a customary criterion to guide how to approach a given bioactivity modeling task from an algorithmic standpoint and how to validate the resulting models.

AUTHOR INFORMATION

Corresponding Author

*E-mail: isidrolauscher@gmail.com.

Author Contributions

I.C.C. designed the study. I.C.C. trained the models, analyzed the results, and prepared the figures. I.C.C., A.B., and T.M. wrote the paper.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

I.C.C. is a fellow of the Paris-Pasteur International PhD Programme. I.C.C. and T.M. thank CNRS and Institut Pasteur for funding. A.B. thanks the European Research Commission (Starting Grant ERC-2013-StG 336159 MIXTURE) for funding. The authors acknowledge the computational resources provided by the Institut Pasteur.

REFERENCES

(1) Cortes Ciriano, I.; Ain, Q. U.; Subramanian, V.; Lenselink, E. B.; Mendez Lucio, O.; IJzerman, A. P.; Wohlfahrt, G.; Prusis, P.; Malliavin, T.; van Westen, G.; Bender, A. Polypharmacology modelling using proteochemometrics: recent developments and future prospects. *MedChemComm* **2015**, *6*, 24−50.

(2) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2011**, *40*, D1100−D1107.

(3) Wang, Y.; Xiao, J.; Suzek, T. O.; Zhang, J.; Wang, J.; Zhou, Z.; Han, L.; Karapetyan, K.; Dracheva, S.; Shoemaker, B. A.; Bolton, E.; Gindulyte, A.; Bryant, S. H. PubChem's BioAssay database. *Nucleic Acids Res.* **2012**, *40*, 400−412.

(4) Fuller, W. A. *Measurement error models*; John Wiley & Sons, Inc.: New York, 2008; pp 441−445.

(5) Wenlock, M. C.; Carlsson, L. A study of how experimental errors influence DMPK QSAR/QSPR models. *J. Chem. Inf. Model.* **2015**, *55*, 125−134.

(6) Kramer, C.; Kalliokoski, T.; Gedeck, P.; Vulpetti, A. The experimental uncertainty of heterogeneous public Ki data. *J. Med. Chem.* **2012**, *55*, 5165−5173.

(7) Kalliokoski, T.; Kramer, C.; Vulpetti, A.; Gedeck, P. Comparability of mixed IC50 data - A statistical analysis. *PloS One* **2013**, *8*, e61007.

(8) Cortes-Ciriano, I.; Bender, A.; Malliavin, T. E. Prediction of PARP inhibition with proteochemometric modelling and conformal prediction. *Mol. Inf.* **2015**, DOI: 10.1002/minf.201400165.

(9) Cortes-Ciriano, I.; Murrell, D. S.; van Westen, G.; Bender, A.; Malliavin, T. Prediction of the potency of mammalian cyclooxygenase inhibitors with ensemble proteochemometric modeling. *J. Cheminf.* **2014**, *7*, 1.

(10) Cortes Ciriano, I.; van Westen, G.; Lenselink, E. B.; Murrell, D. S.; Bender, A.; Malliavin, T. Proteochemometrics modeling in a Bayesian framework. *J. Cheminf.* **2014**, *6*, 35.

(11) Brown, S. P.; Muchmore, S. W.; Hajduk, P. J. Healthy skepticism: assessing realistic model performance. *Drug Discovery Today* **2009**, *14*, 420−427.

(12) Tsang, S.; Kao, B.; Yip, K.; Ho, W.-S.; Lee, S.-D. Decision trees for uncertain data. *IEEE Trans. Knowl. Data Eng.* **2009**, *23*, 64−78.

(13) Ge, J.; Xia, Y.; Tu, Y. A discretization algorithm for uncertain data. *Database and Expert Systems Applications*; 2010; Vol. *6262*, pp 485−499.

(14) Qin, B.; Xia, Y.; Li, F. *A Bayesian classifier for uncertain data*, Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22−26, 2010; 2010.

(15) Zhang, J. B. T. Support vector classification with input data uncertainty. *Adv. Neural Inf. Process. Syst* **2004**, *17*, 161−169.

(16) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for machine learning*; MIT Press: 2006.

(17) Atla, A.; Tada, R.; Sheng, V.; Singireddy, N. Sensitivity of different machine learning algorithms to noise. *J. Comput. Sci. Coll.* **2011**, *26*, 96−103.

(18) Hawkins, D. M.; Basak, S. C.; Mills, D. Assessing model fit by cross-validation. *J. Chem. Inf. Model.* **2003**, *43*, 579−586.

(19) Zhu, X.; Wu, X.; Chen, Q. Eliminating class noise in large data sets; 2003. http://www.aaai.org/Papers/ICML/2003/ICML03-119.pdf (accessed June 1, 2015).

(20) Zhu, X.; Wu, X. Class noise vs. attribute noise: a quantitative study. *Artif. Int. Rev.* **2004**, *22*, 177−210.

(21) Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; Tewari, A. *Adv. Neural. Inf. Process. Syst.*; Curran Associates, Inc.: 2013; pp 1196−1204.

(22) Machine Learning Algorithms: A study on noise sensitivity. In *Proc. 1st Balcan Conference in Informatics 2003, pp 356-365, Thessaloniki, November 2003*, Manolopoulos, Y., Spirakis, P., Eds.; 2003.

(23) Teytaud, O. *Robust learning: regression noise*; 2001.

(24) Kearns, M. Efficient noise-tolerant learning from statistical queries. *JACM* **1998**, *45*, 983−1006.

(25) Angluin, D.; Laird, P. Learning from noisy examples. *Mach. Learn.* **1988**, *2*, 343−370.

(26) Nettleton, D.; Orriols-Puig, A.; Fornells, A. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif. Int. Rev.* **2010**, *33*, 275−306.

(27) John, G. H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. 1995. http://dl.acm.org/citation.cfm?id/2074158.2074196 (accessed June 1, 2015).

(28) Quinlan, J. R. *C4.5: programs for machine learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.

(29) Aha, D. W.; Kibler, D.; Albert, M. K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37−66.

(30) Platt, J. C. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*; MIT Press: Cambridge, MA, 1999; pp 185−208.

(31) Norinder, U.; Bostrom, H. Introducing uncertainty in predictive modeling-friend or foe? *J. Chem. Inf. Model.* **2012**, *52*, 2815−2822.

(32) Steinwart, I. On the influence of the kernel on the consistency of Support Vector Machines. *J. Mach. Learn. Res.* **2002**, *2*, 67−93.

(33) Tipping, M. E. Sparse Bayesian learning and the Relevance Vector Machine. *J. Mach. Learn. Res.* **2001**, *1*, 211−244.

(34) Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273−297.

(35) Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and regression trees*; Wadsworth and Brooks: Monterey, CA, 1984.

(36) Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **2000**, *29*, 1189−1232.

(37) Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobot.* **2013**, *7*, 21.

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

(39) Wold, S.; Sjöström, M.; Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemom. Intell. Lab. Syst.* **2001**, *58*, 109−130.

(40) Fix, E.; Hodges, J. L. An important contribution to nonparametric discriminant analysis and density estimation: Commentary on Fix and Hodges (1951). *Int. Stat. Rev.* **1989**, *57*, 233−247.

(41) Paricharak, S.; Cortes-Ciriano, I.; IJzerman, A. P.; Malliavin, T. E.; Bender, A. Proteochemometric modeling coupled to in silico target prediction: an integrated approach for the simultaneous prediction of polypharmacology and binding affinity of small molecules. *J. Cheminf.* **2015**, *7*, 15.

(42) Chen, H.; Carlsson, L.; Eriksson, M.; Varkonyi, P.; Norinder, U.; Nilsson, I. Beyond the scope of Free-Wilson analysis: building interpretable QSAR models with machine learning algorithms. *J. Chem. Inf. Model.* **2013**, *53*, 1324−1336.

(43) Murrell, D. S.; Cortes-Ciriano, I.; van Westen, G.; Malliavin, T.; Bender, A. Chemistry Aware Model Builder (camb): an R package for predictive bioactivity modeling. 2014. http://github.com/cambDI/camb (accessed June 1, 2015).

(44) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(45) Glem, R. C.; Bender, A.; Arnby, C. H.; Carlsson, L.; Boyer, S.; Smith, J. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. *IDrugs* **2006**, *9*, 199−204.

(46) Landrum, G. RDKit Open-source cheminformatics. 2006. http://rdkit.org/ (accessed June 1, 2015)..

(47) Cortes-Ciriano, I. FingerprintCalculator. 2014. http://github.com/isidroc/FingerprintCalculator (accessed June 1, 2015).

(48) Bender, A.; Jenkins, J. L.; Scheiber, J.; Sukuru, S. C. K.; Glick, M.; Davies, J. W. How similar are similarity searching methods? A principal component analysis of molecular descriptor space. *J. Chem. Inf. Model.* **2009**, *49*, 108−119.

(49) Koutsoukas, A.; Paricharak, S.; Galloway, W. R. J. D.; Spring, D. R.; IJzerman, A. P.; Glen, R. C.; Marcus, D.; Bender, A. How diverse are diversity assessment methods? A comparative analysis and benchmarking of molecular descriptor space. *J. Chem. Inf. Model.* **2014**, *54*, 230−242.

(50) Kuhn, M. Building predictive models in R using the caret package. *J. Stat. Soft* **2008**, *28*, 1−26.

(51) Kuhn, M.; Johnson, K. *Applied predictive modeling*; Springer New York: New York, NY, 2013.

(52) Genton, M. G. Classes of kernels for machine learning: a statistics perspective. *J. Mach. Learn. Res.* **2002**, *2*, 299−312.

(53) *Kernel methods in computational biology*; Schölkopf, B., Tsuda, K., Vert, J., Eds.; MIT Press: 2004.

(54) MacKay, D. J. C. *Information theory, inference and learning algorithms*; Cambridge University Press: 2003.

(55) Puntanen, S.; Styan, G. P. H. In *The Schur complement and its applications*; Zhang, F., Ed.; Numerical Methods and Algorithms 4; Springer: US, 2005; pp 163−226.

(56) Ben-Hur, A.; Ong, C. S.; Sonnenburg, S.; Schölkopf, B.; Rätsch, G. Support Vector Machines and kernels for computational biology. *PLoS Comput. Biol.* **2008**, *4*, e1000173.

(57) Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning*; Springer Series in Statistics; Springer New York Inc.: New York, NY, USA, 2001.

(58) Breiman, L. Arcing classifier (with discussion and a rejoinder by the author). *Ann. Statist.* **1998**, *26*, 801−849.

(59) Sheridan, R. P. Using Random Forest to model the domain applicability of another Random Forest model. *J. Chem. Inf. Model.* **2013**, *53*, 2837−2850.

(60) Sheridan, R. P. Three useful dimensions for domain applicability in QSAR models using random forest. *J. Chem. Inf. Model.* **2012**, *52*, 814−823.

(61) Abdi, H. Partial least squares regression and projection on latent structure regression (PLS Regression). *Wiley Interdiscip. Rev.: Comput. Statistics* **2010**, *2*, 97−106.

(62) Abdi, H.; Williams, L. J. Principal component analysis. *Wiley Interdiscip. Rev.: Comput. Statistics* **2010**, *2*, 433−459.

(63) Karatzoglou, A.; Smola, A.; Hornik, K.; Zeileis, A. kernlab - An S4 package for kernel methods in R. *J. Stat. Soft.* **2004**, *11*, 1−20.

(64) Ridgeway, G. *gbm: generalized boosted regression models*; R package version 2.1; 2013.

(65) Venables, W. N.; Ripley, B. D. *Modern applied statistics with S*, 4th ed.; Springer: New York, 2002.

(66) Mevik, B.-H.; Wehrens, R.; Liland, K. H. *pls: partial least squares and principal component regression*; R package version 2.4-3; 2013.

(67) Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18−22.

(68) Peters, A. Hothorn, T. *ipred: improved predictors*; R package version 0.9-3; 2013.

(69) Winer, B.; Brown, D.; Michels, K. *Statistical principles in experimental design*; McGraw-Hill series in psychology, 3rd ed.; McGraw-Hill: New York, 1991.

(70) Efron, B.; Tibshirani, R. *An introduction to the bootstrap*; Chapman & Hall: New York, 1993.

(71) Long, P. M.; Servedio, R. A. Random classification noise defeats all convex potential boosters. *Mach. Learn.* **2010**, *78*, 287−304.

(72) Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Mach. Learn.* **2000**, *40*, 139−157.

(73) Kononenko, I.; Kukar, M. *Machine learning and data mining: introduction to principles and algorithms*; Horwood Publishing Limited: 2007.

(74) Wu, Y.; Ianakiev, K.; Govindaraju, V. Improved k-nearest neighbor classification. *Pattern Recognit.* **2002**, *35*, 2311−2318.

(75) Sánchez, J. A.; Luengo, J.; Herrera, F. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognit.* **2013**, *46*, 355−364.

(76) Sánchez, J. A.; Luengo, J.; Herrera, F. *Hybrid Artificial Intelligence Systems*; Lecture Notes in Computer Science; Springer International Publishing: 2014; Vol. *8480*, pp 597−606.

(77) Everitt, B. S.; Landau, S.; Leese, M.; Stahl, D. *Cluster analysis*; John Wiley & Sons, Ltd.: 2011; pp 215−255.

(78) Golbraikh, A.; Tropsha, A. Beware of q2! *J. Mol. Graphics Modell.* **2002**, *20*, 269−276.

(79) Kramer, C.; Lewis, R. QSARs, Data and Error in the Modern Age of Drug Discovery. *Curr. Top. Med. Chem.* **2012**, *12*, 1896−1902.