Journal of Cheminformatics

RESEARCH ARTICLE

Open Access

The effect of noise on the predictive limit of QSAR models

Scott S. Kolmar* and Christopher M. Grulke

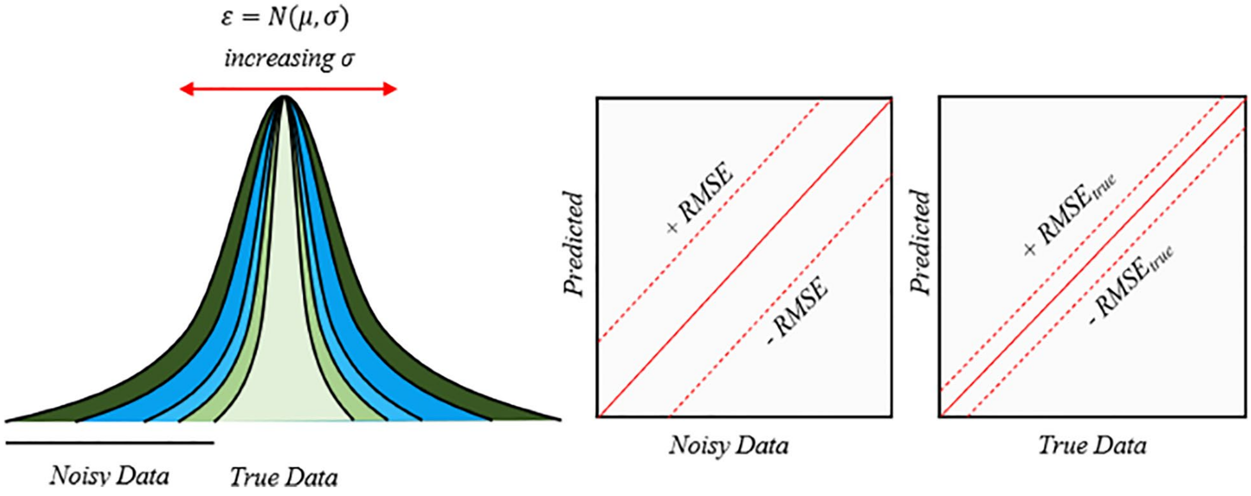
Abstract

A key challenge in the field of Quantitative Structure Activity Relationships (QSAR) is how to effectively treat experimental error in the training and evaluation of computational models. It is often assumed in the field of QSAR that models cannot produce predictions which are more accurate than their training data. Additionally, it is implicitly assumed, by necessity, that data points in test sets or validation sets do not contain error, and that each data point is a population mean. This work proposes the hypothesis that QSAR models *can* make predictions which are more accurate than their training data and that the error-free test set assumption leads to a significant misevaluation of model performance. This work used 8 datasets with six different common QSAR endpoints, because different endpoints should have different amounts of experimental error associated with varying complexity of the measurements. Up to 15 levels of simulated Gaussian distributed random error was added to the datasets, and models were built on the error laden datasets using five different algorithms. The models were trained on the error laden data, evaluated on *error-laden* test sets, and evaluated on *error-free* test sets. The results show that for each level of added error, the RMSE for evaluation on the error free test sets was always better. The results support the hypothesis that, at least under the conditions of Gaussian distributed random error, QSAR models can make predictions which are more accurate than their training data, and that the evaluation of models on error laden test and validation sets may give a flawed measure of model performance. These results have implications for how QSAR models are evaluated, especially for disciplines where experimental error is very large, such as in computational toxicology.

Keywords: Error, Prediction error, Model evaluation, Gaussian process

*Correspondence: Kolmar.Scott@epa.gov
Center for Computational Toxicology and Exposure, Office of Research and Development, US Environmental Protection Agency, Research Triangle Park, NC, USA

Graphical Abstract



Introduction

One of the key challenges in Quantitative Structure Activity Relationship (QSAR) modeling is evaluating the predictive performance of models, and evaluation methodology has been the subject of many studies in the past several decades [1–6]. Evaluation of predictive performance has critical implications for the fields of drug discovery [6, 7], toxicological risk assessment [8], and environmental regulation [9], among others. The importance of model evaluation and comparison is reflected in the fourth principle from the Organization for Economic Cooperation and Development (OECD), which states that a QSAR model must have "appropriate measures of goodness of fit, robustness, and predictivity" [9, 10]. While best practice guidelines have often emphasized the need for external validation on compounds that have been rigorously excluded from the training set, implicit assumptions about error in the training and validation data, and how these assumptions might affect performance evaluation, tend to be overlooked [1–3]. It is necessary to examine these assumptions and their effects in order to appropriately evaluate the predictivity of QSAR models and utilize their predictions with confidence.

The most problematic assumption about errors implicitly made during most QSAR modeling is that the given value for any experimental endpoint is the "true" value for that measurement. This assumption is necessarily taken when the following conditions are met: when the endpoint values are represented as single measurements, and when models are compared via their prediction metrics, such as root mean squared error (RMSE) and the coefficient of determination (R^2). As detailed below, it is often the case that endpoint values are represented as single measurements, and this obligates the modeler to assume that these measurements are representative of the true value. Additionally, the modeler must then compare models using performance metrics that implicitly assume endpoint quantities to be sufficiently representative of physical truth, that is, there is no mathematical mechanism built in to account for the fact that these single values may be several standard deviations away from the actual population mean of that measurement. To put all of this in more rigorous statistical terms, the assumption is made that the given experimental value is the sample mean, and that this sample mean sufficiently approximates the population mean (true value) of all possible measurements [11]. This assumption is made for two main reasons. The first reason is that most models are built on datasets which have only a single, or at best three replicates, for any given measurement and therefore the data does not support a more detailed understanding of the population distribution and uncertainties. For example, an analysis performed on a large set of drug metabolism and pharmacokinetic (DMPK) data showed that 87% of the 358,523 measurements had only a single replicate [7]. Second, most machine learning algorithms, with the exception of Bayesian methods such as Gaussian Process [12] and conformal prediction [13, 14], treat endpoints as discrete quantities rather than distributions thereby forcing QSAR modelers to use only a single value when applying most learning methods. Unfortunately, the assumption that the single experimental value is a good representative of the population mean is often not true. It is unlikely that a measurement's sample mean will closely approximate the population mean unless the number of replicates is very high [11], although for endpoints which involve fitting a curve to measurements made at multiple concentrations, parametric bootstrapping can provide a workaround to the issue of having few replicates [15]. In sum, the assumption that experimental endpoints are true values ignores the reality that experimental measurements have a distribution and uncertainty associated with them, and this statistical reality has important effects on the predictivity of QSAR models.

Ignoring experimental error of the target property creates two main problems in modeling studies. The first issue is that inaccurate training data may cause a QSAR model to fit the trends in the noise rather than the underlying trends in the data, a well-known phenomenon called overfitting [16]. Overfitting can be diagnosed because performance metrics such as root mean squared error (RMSE) and the coefficient of determination (R^2) will be far worse for the test set than for the training set [16]. The second and more pernicious issue is that endpoint values in the test set also have experimental error, but these test set values set the standard by which a model is evaluated. If these error laden test set values are used to calculate a model's performance statistics, then even if a QSAR model predicts close to the true value, the error for that prediction will be observed as high if the experimental test set value is far from the true value (Fig. 1).

Experimental measurements are complicated by two main sources of error. Systematic error biases a measurement in one direction and can be the result of natural or instrumental phenomena [11]. Random error, by definition, is equally likely to affect a measurement in either direction. Systematic error is notoriously difficult, if not impossible, to identify statistically, but random error can be treated effectively using a Gaussian distribution [17]. Experimental error, in the absence of known systematic error, is generally treated to be random. This contention is well supported because variability in natural processes tends to be random, due to contributions from many small underlying factors. The central limit theorem allows us to model random processes using the Gaussian distribution [18]. The sum of this line of reasoning is that most experimental error, in the absence of identifiable systematic error, can be reasonably modeled using a Gaussian distribution. Furthermore, there is a verifiable body of scientific literature (especially in chemistry) which shows that measurements tend to an average value with a Gaussian distribution. [11, 19, 20].

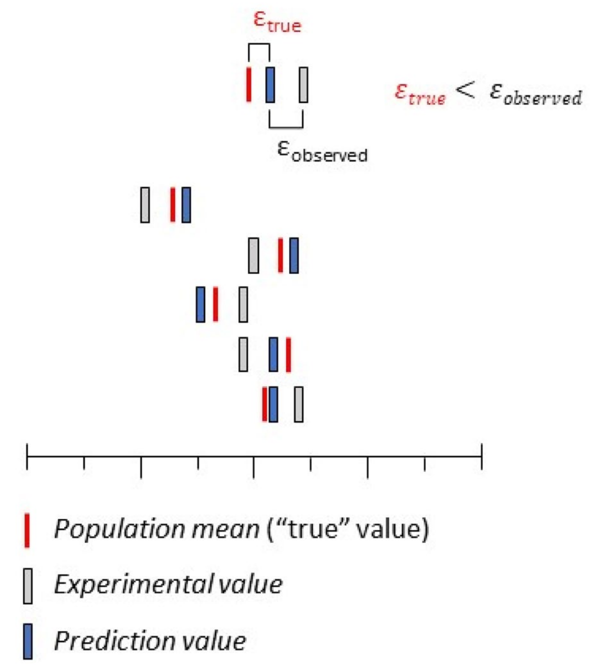


Fig. 1 Graphical representation of experimental error and prediction error for an arbitrary dataset. Each row of bars is a separate observation for an arbitrary endpoint. The red bars represent population means ("true" values) for the observation, the grey bars represent experimental values, and the blue bars represent prediction values. If prediction values are closer to the population means than the experimental values, the true error will be smaller than the observed error

Drawing from this accepted treatment of experimental error, several studies have attempted to better understand the relationship between random experimental error and predictivity. Several works have analyzed proprietary pharmaceutical data and public databases in order to estimate the average error in commonly measured pharmacological and toxicological quantities such as pK_i, [20] pIC_{50}, [19] and cytotoxicity [21]. Brown and coworkers used a computational approach to develop empirical rules for distributions of coefficients of determination (R^2) based on dataset parameters such as range of endpoint values and number of samples [6]. All three of these studies provide benchmarks to evaluate whether or not the predictivity of any given model is reasonable or not, given what is known about average random error in commonly measured endpoints and how this error propagates to performance statistics such as RMSE and R^2. A seminal contribution to this topic comes from Cortes-Ciriano and coworkers, in which they performed a full factorial study of random experimental error on 12 different datasets, 12 algorithms, and 10 levels of simulated random experimental error [22]. The results showed that algorithms have different levels of sensitivity to added random experimental error, such that while algorithm A might have a lower RMSE than algorithm B at low noise levels, algorithm A can have a higher RMSE than algorithm B at high noise levels.

A common assertion in the QSAR literature, which is brought up in the studies mentioned above, is that the experimental error in a dataset puts a hard limit on the predictivity of a model, or in other words, that a model cannot make predictions which are more accurate than its training data [7, 20, 23]. The assertion that there is a hard limit on prediction accuracy relies on the assumption that the test set values are true values, but as mentioned above, the test set values also have experimental error. This work poses the main hypothesis that a QSAR model can indeed make predictions which are more accurate than the training data; however, we are unable to validate that these models are better than our test data. This hypothesis is made under the condition that the experimental error is Gaussian distributed; this condition is certainly not representative of every real-world data situation, but it allows the hypothesis to be tested under a set of ideal conditions. A logical method of testing this hypothesis is to compare $\mathrm{RMSE}_{\mathrm{observed}}$ and $\mathrm{RMSE}_{\mathrm{true}}$ for a variety of models, which requires model development with two sets of values for each molecule in a dataset, artificially generated error laden experimental values and true values.

Understanding the effect of error on predictivity is particularly important for the field of toxicology, because environmental risk assessments and subsequent regulations depend on the results and confidence intervals of the predictions [24]. Furthermore, toxicological models are often built on in vivo or animal studies measurements, which are notoriously variable due to the myriad factors which contribute to error [8, 25]. It has been posited that variance in the experimental data contributes more to prediction error than the error from the model itself [26, 27]. If the hypothesis that a QSAR model can make predictions which are more accurate than the training data is true, then it would suggest that models trained on highly variable toxicological datasets could produce accurate and therefore reliable predictions. This work will test this hypothesis and discuss the results in the context of toxicological datasets.

Methods

Experimental design

Residual error in a model prediction for a validation compound can be understood in two different ways. Based on the assumption that the experimentally measured values are true, the error is calculated as simply the difference between the observed value and the predicted value, $\varepsilon_{\mathrm{observed}}$. However, the error of interest for a predictive model is actually the difference between the population mean and the predictive value, $\varepsilon_{\mathrm{true}}$ (Fig. 1). While this argument conforms with our understood goals for a QSAR model, population means are difficult to ascertain for most endpoints of biological relevance, and therefore, $\varepsilon_{\mathrm{true}}$ is often out of reach. However, if a computational experiment made it possible to determine $\varepsilon_{\mathrm{true}}$, it would allow us to investigate the question of whether there is a hard limit on the predictivity of a model, or if the limit is actually on our ability to accurately measure the predictivity of the model.

If Y is the vector of experimental endpoints, Z the vector of true values, and $\hat{Y}$ the vector of model predictions, then ε_i is the difference between an experimental measurement and the true value, $RMSE_{observed}$ is the prediction error calculated from the experimental endpoints, and $RMSE_{true}$ is the prediction error calculated from the true values. $RMSE_{observed}$ will be higher than $RMSE_{true}$ if the average ε is large. The problematic assumption is that ε is assumed to be 0. This means that $RMSE_{observed}$ is often mistaken for $RMSE_{true}$ when evaluating a QSAR model, and thus the true predictivity of a model is probably underestimated.

$$Y = \left(y_1, \ldots, y_k\right) \tag{1}$$

$$Z = (z_1, \ldots, z_k) \tag{2}$$

$$\hat{\mathrm{Y}} = \left(\hat{\mathrm{y}}_1, \ldots, \hat{\mathrm{y}}_k\right) \tag{3}$$

$$\varepsilon_i = \left(y_i - z_i\right) \tag{4}$$

$$RMSE_{observed} = \sqrt{\frac{\sum_{i=1}^{k}(\hat{\mathrm{y}}_i - y_i)^2}{k}} \tag{5}$$

$$RMSE_{true} = \sqrt{\frac{\sum_{i=1}^{k}(\hat{\mathrm{y}}_i - z_i)^2}{k}} \tag{6}$$

As mentioned above though, evaluating this claim requires having both true values and experimental measurements to assess whether $\mathrm{RMSE}_{\mathrm{observed}}$ is greater than $\mathrm{RMSE}_{\mathrm{true}}$. In the absence of such true values, we assume that for the datasets used in this experiment, the original values are the true values and then create simulated "experimental" observations adding increasing amounts of Gaussian error to those "true" values. This assumption (which directly contradicts our premise that doing so is dangerous) and its possible ramifications will be addressed in the "Data sets" and "Discussion and conclusion" sections.

While the effect of error (including the addition of simulated error) has been studied in previous publications by Cortes-Ciriano and co-workers [22], this work has some key differences which are important to highlight. The purpose of the previous work was to systematically explore how different algorithms respond to simulated error in order to benchmark performance. The authors achieved this by modeling 12 different protein pIC_{50} datasets with 12 algorithms, and by observing the increase in RMSE as simulated Gaussian distributed error is added to these datasets. While the datasets showed a diversity of targets, the type of endpoint, pIC_{50}, is the same for each dataset. Additionally, the range of estimated native experimental error for these datasets is only 1.1 log units. In contrast, the objective of the present work is to test the hypothesis that a QSAR model can predict more accurately than the dataset on which it is trained. The present approach, similarly, is to use several common algorithms to model different datasets and observe how the addition of simulated Gaussian distributed error affects the RMSE. A key difference here, however, is to compare the performance statistics for a model's prediction on the noisy data vs a model's prediction on the true data in an effort to de-couple the potential causes of observed prediction error and assess their individual impacts on our observed model performance. The hope is to be able to separate and better understand three potential causes of error: learning error (i.e., prediction error caused by the modeling methodology being insufficient), propagated training set error (i.e., prediction error caused by the training set having errors that are then learned by the model), and validation error (i.e., prediction "error" that is perceived due to the validation set itself having error). Here, $RMSE_0$ can be understood as learning error; $RMSE_{true}$—$RMSE_0$ would approximate propagated training error; and $RMSE$—$RMSE_{true}$ would approximate the validation error.

Data sets

All data sets have error associated with their target properties, including random experimental error and systematic error. However, the magnitude of random experimental error and the nature of systematic error varies widely with the type of endpoint. For this study, we primarily consider the differences in error characteristics likely within five categories of endpoints: quantum mechanical calculations, physicochemical properties, biochemical binding, in vitro bioactivity, and in vivo toxicity. While quantum mechanical calculations do not have random experimental error, because the same calculation will give exactly the same number [28], systematic error is prevalent and comes from the fundamental choice of exchange–correlation approximations used in the density functional theory (DFT) method [29]. Measurements of physiochemical properties typically require determination of equilibrium concentrations of compounds in various solvents or phases [30], and are often made with standard analytical chemistry methods such as liquid chromatography, gas chromatography, or spectroscopy [31]. The random experimental and systematic error associated with these measurements thus comes from factors such as the purity of the compounds, instrument calibration, and instrument sensitivity [11]. In contrast, factors in biochemical measurements, such as protein purity, accurate determination of protein concentration, and equilibration time contribute to higher random experimental and systematic error that can make these measurements highly variable [32]. Toxicological datasets can include many different types of in vitro bioactivity and in vivo measurements, which are sometimes aggregated in order to provide composite scores for use in classification problems [33, 34]. These datasets likely have the highest level of random experimental and systematic error because the sources of error are diverse and the accumulated errors propagate. Utilizing datasets from each of these categories allows a comparison to be made between datasets with which are likely to have increasing amounts of native random experimental error thereby allowing us to investigate how our assumption regarding the "truth" of the values provided in the dataset affects our conclusions.

The majority of quantum mechanical [35, 36], physiochemical [38], and biochemical data [39] sets included in this analysis were taken from MoleculeNet [40], a large curated collection of chemical data which is intended to provide standard benchmarking data sets for QSAR models. As the primary goal of this work is to benchmark different common QSAR algorithms, the MoleculeNet collection provides several high-quality data sets for comparison. In vitro bioactivity sets were obtained from the EPA's ToxCast [34] database and in vivo toxicity datasets were represented by an LD_{50} data set gathered a report by Gadeleta and coworkers; 75% of these LD_{50} values were taken from the EPA's DSSTox database, with the other 25% assembled from literature publications as described in Gadeleta et al. [41] A summarization of the datasets used is available in Table 1. Additional dataset details are described in Additional file 1.

Descriptors

Molecular descriptors were generated using PadelPy [42], a python package which wraps the Padel descriptor software [43, 44], or with OPERA, an open source software package which also generates Padel descriptors [45, 46]. The Padel software generates up to 1,875 descriptors, including 1444 1D/2D, and 431 3D descriptors. These quantities include electrotopological, topochemical, and linear free energy descriptors, as well as ring counts, McGowan volume, Crippen's logP, and others. While there are many choices of descriptor sets [4, 47, 48], Padel descriptors are commonly used in QSAR workflows and thus provide a logical and reasonable method

Table 1 Datasets used in this work, with the number of molecules, endpoint, endpoint units, range, and reference for each

Dataset	Category	Entries[a]	Endpoint	Range	Refs.
G298_atom	Quantum mechanical	131,082	ΔG^{o}_{at} (kcal mol^{-1})	− 2417to − 288	[29, 30]
Alpha	Quantum mechanical	131,082	α ($Bohr^3$)	9.0 to 27.8	[29, 30]
Lip	Physiochemical	4200	logD	− 1.5 to 4.5	[31]
Solv	Physiochemical	642	ΔG^{o}_{hyd} (kcal mol^{-1})	− 25.5 to 3.4	[32]
BACE	Biochemical	1513	pIC_{50}	2.5 to 10.5	[33]
Tox_102[b]	Toxicological in vitro	971	$logAC_{50}$	− 2.1 to 4.7	[28]
Tox_134[c]	Toxicological in vitro	1347	$logAC_{50}$	− 4.0 to 2.8	[28]
LD_{50}	Toxicological in vivo	5003	$logLD_{50}$ (mg kg^{-1})	− 1.9 to 4.8	[35]

[a] Original size of the dataset. If datasets have more than 1000 molecules, they were randomly sampled down to a size of 1000 before modeling

[b] Includes data exclusively from the ATG-PPre-cis assay

[c] Includes data exclusively from the ATG-PPARg-trans assay

for performing a proof of concept study such as the work presented here. For this work, only the 1,444 1D and 2D Padel descriptors were used.

Modeling workflow

Padel descriptors were first generated using PadelPy or OPERA using a SMILES string for each molecule [49]. In some cases, a subset of descriptors (up to 12) had infinite values, in which case those descriptors were removed from the dataset. If the dataset has more than 1000 molecules, it was sampled down to a size of 1,000; if the dataset has less than 1000 molecules, it continued without sampling. Custom code was written in python, utilizing the popular machine learning package scikit-learn [50], to run the dataset through a machine learning pipeline. The code implemented the following workflow on each dataset. The endpoint data column was used to generate 15 additional endpoint data columns with increasing levels of gaussian distributed noise. This process was repeated five times at each noise level to give 75 total datasets. A machine learning algorithm was chosen, such as k-nearest neighbors (kNN) or random forest (RF). The algorithm was then preprocessed, optimized, trained, and fit on each of the 75 datasets with added noise, giving 75 unique models, 75 RMSE's, and 75 R^2's. Each of these models was then fit on the original dataset which has no added noise, giving an additional 75 $RMSE_{true}$'s and 75 R^2_{true}'s. The RMSE, $RMSE_{true}$, R^2, and R^2_{true} values were then plotted against noise level. This process was repeated for each algorithm and each dataset. Details on each step of this process are given below. A graphical representation of the modeling workflow and machine learning pipeline is provided in Fig. 2.

Machine learning pipeline

Prior to modeling a given dataset, 25% of the data was split into a test set, and 75% of the data was split into a training set. Each algorithm was then put through a pipeline of steps before training on the training set and predicting on the test set. This pipeline consisted of three steps: scaling, principal component analysis (PCA), and algorithm fitting, with PCA and algorithm hyperparameters optimized using fivefold GridSearchCV or RandomSearchCV. Scaling was applied to all features (descriptor values) using StandardScaler, which centers each feature on the mean and scales to unit variance, which is a common requirement for many algorithms. Dimension reduction was then applied using PCA, optimizing the number of principal components. Algorithm hyperparameters were then optimized as shown in Table 2.

Random error generation

Random error was added to datasets by sampling from a Gaussian distribution of zero mean and increasing standard deviation σ_{noise}. Noise was added only to the target variables and not to the descriptors. This σ_{noise} was determined from the product of the range of endpoint values in the dataset, the noise level *n,* and a multiplier. This multiplier was set to 0.01 after experimentation with a range of values and observing the effect on RMSE. Each dataset was used to generate 15 noise levels with 5 replicates at each noise level. Because *n* starts at 0, the 0th noise level has no added noise.

$$Y_{noise_n,i} = Y + N\left(0, \sigma_{noise_n}\right) \tag{7}$$

$$\sigma_{noise_n} = (Y_{max} - Y_{min})*multiplier*n \tag{8}$$

$$n \in (0, \ldots, 14)$$

$$i \in (1, \ldots, 5)$$

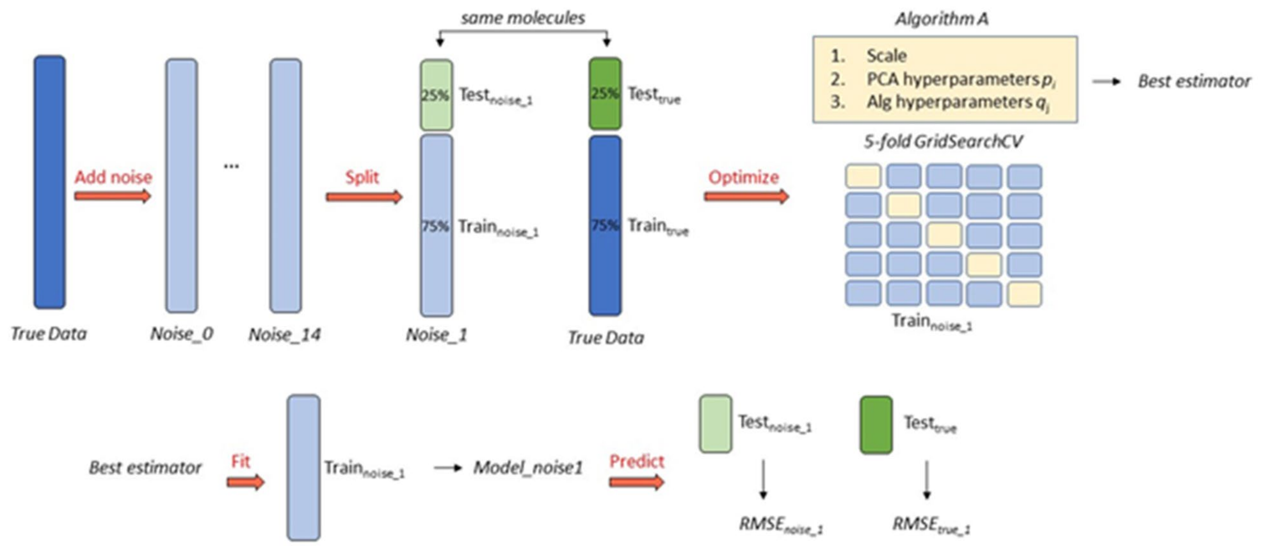


Fig. 2 The modeling workflow and machine learning pipeline which is used in this work. Fifteen noisy datasets are first generated by adding Gaussian distributed noise to the dataset. Each of these noisy datasets is then split into $\text{Test}_{\text{noise\_n}}$ and $\text{Train}_{\text{noise\_n}}$, and the true data is split into $\text{Test}_{\text{true}}$ and $\text{Train}_{\text{true}}$ (no training is performed on $\text{Train}_{\text{true}}$). The hyperparameters for PCA and the chosen algorithm are then optimized using GridSearchCV (or RandomSearchCV) to give a *Best estimator*. The *Best estimator* is then fit on $\text{Train}_{\text{noise\_n}}$ to give *Model_noise*$_n$, and *Model_noise*$_n$ predicts on both $\text{Test}_{\text{noise\_n}}$ and $\text{Test}_{\text{true}}$ to give $\text{RMSE}_{\text{noise\_n}}$ and $\text{RMSE}_{\text{true\_n}}$. This procedure is repeated for each noisy dataset

Table 2 Algorithms used in this work and their respective hyperparameter optimization spaces

Algorithm	Hyperparameters searched in optimization[a,b]
Ridge regression (Ridge)	*PCA n components* $\in (1, 3, \ldots, 59)$ $\alpha \in (1, 2, 3, 4, 5, 10)$
k-nearest neighbors (kNN)	*PCA n components* $\in (1, 3, \ldots, 59)$ $k \in (1, 2, \ldots, 20)$
Support vector regressor (SVR)	*PCA n components* $\in (1, 3, \ldots, 59)$ $C \in (0.01, 0.1, 1, 10)$ *kernel:* radial basis function (RBF)
Random forest (RF)	*PCA n components* $\in (1, 3, \ldots, 59)$ *n estimators* $\in (1, 10, \ldots, 200)$ *max depth* $\in (1, 3, \ldots, 99)$ *max leaf nodes* $\in (2, 12, \ldots, 92)$
Gaussian process (GP)	*PCA n components* $\in (1, 3, \ldots, 59)$ *kernel:*[c] RBF, WhiteKernel, Matern, DotProduct, ExpSineSquared, ConstantKernel or RationalQuadratic *Normalize y*: true

[a] Ridge, kNN, SVR, and GP algorithms were optimized using fivefold *GridSearchCV*, but RF was optimized using fivefold *RandomSearchCV* with 500 iterations

[b] All algorithm hyperparameters which are not listed in this table were set to the defaults provided in the sci-kit learn library

[c] For most datasets, only a single kernel converged. So the kernel was not optimized in GridSearchCV, it was chosen beforehand and used for the entire dataset

Algorithms Several machine learning algorithms were chosen for this study which are common to QSAR modeling workflows and which represent a variety of mathematical approaches for capturing complex patterns in data. Applying this analysis to a selection of algorithms allows us to determine whether the ability to make predictions which are more accurate than the training data is conserved across a variety of methods.

Ridge regression

Ridge regression is a form of linear regression that utilizes a technique called regularization in order to reduce model complexity and minimize overfitting [51]. If a dataset has n number of features x, then a linear model calculates a prediction $\hat{y}$ as a function of n number of weight coefficients β times x (Eq. 9). The resulting cost function for this linear model simply minimizes the squared difference between predictions $\hat{y}$ and observations y by adjusting β (Eq. 10). Ridge regression adds a regularization term to this cost function which contains a regularization coefficient λ times the square of each weight coefficient β (Eq. 11). This λ is set as a hyperparameter for the ridge regression algorithm. The larger λ is, the more a particular coefficient will be dampened by the cost function. This means that if some feature x_i is dominating the linear model, causing overfitting, the weight coefficient will be dampened and the influence on the model will be reduced.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{9}$$

$$L = \sum_{i=1}^{M} (\hat{y} - y)^2 = \sum_{i=1}^{M} \left(\hat{y} - \sum_{j=0}^{n} \beta_j x_{ij}\right)^2 \tag{10}$$

$$L = \sum_{i=1}^{M} (\hat{y} - y)^2 = \sum_{i=1}^{M} \left(\hat{y} - \sum_{j=0}^{M} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=0}^{n} \beta_j^2 \tag{11}$$

K nearest neighbors

KNN regression [52] uses distance measures to find the k observations which are closest to the coordinates of the input features in n dimensional vector space. The average observation value of these k neighbors is used to calculate the prediction $\hat{y}$. Each observation y_i of the k nearest neighbors can also be weighted by the distance D_i (Eq. 12). The most common distance measure to use is Euclidean distance (Eq. 13), which was used in this work.

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} D_i y_i \tag{12}$$

$$D = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2} \tag{13}$$

Support vector machines

Support vector machine (SVM) methods [53] are non-parametric algorithms which rely instead on kernel functions to make predictions. SVM's predict complex non-linear trends by transforming the n dimensional input vector space into a higher m dimensional vector space. This is achieved by a mapping function, otherwise known as a kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ which acts on the vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$. Once the input vectors are in the higher dimensional space, a linear hyperplane can be drawn to separate the data by maximizing the margin between each data point and the hyperplane. This hyperplane is a function of the input vector $\boldsymbol{x}$ and the weight vector $\boldsymbol{\beta}$ (Eq. 14). The linear form of this hyperplane can be learned by minimizing the cost function J (Eq. 15). When a kernel function is applied to transform $\boldsymbol{x}$ into a higher dimensional space, and when we define the weighting coefficient vector $\boldsymbol{\beta}$ by a linear combination of the training observations (Eq. 16), we arrive at the new functional form for support vector regressor (SVR) (Eq. 17).

$$\hat{y} = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n = \boldsymbol{\beta} \cdot \boldsymbol{x} \tag{14}$$

$$J(\boldsymbol{\beta}) = \frac{1}{2} \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{\beta} \tag{15}$$

$$\beta = \sum_{i=0}^{n} a_i x_i \tag{16}$$

$$\hat{y} = \sum_{i=0}^{n} a_i y_i K(\boldsymbol{x_i}, \boldsymbol{x}) \tag{17}$$

Random forest

The Random Forest (RF) algorithm [54] is an ensemble method which makes predictions from the average of many individual decision trees predictions. The RF algorithm uses bagging with replacement to create n samples from a dataset and builds a decision tree on each of those bagged samples, creating a "forest" of random decision trees. The features, or input variables x, can also be sampled during this process. This approach reduces the common problem of overfitting with decision trees. In Eq. 18, $T_i(\boldsymbol{x})$ is an individual decision tree trained on a subset of the input variable vector $\mathbf{x}$, and there are B decision trees.

$$\hat{y} = \frac{1}{B} \sum_{i=0}^{B} T_i(\boldsymbol{x}) \tag{18}$$

Results

RMSE response to error

Each dataset was used to generate 15 levels of noise with five replicates at each noise level, and the ridge, kNN, SVR, and RF algorithms were used to model each dataset with the various levels of added noise. These noisy data simulate the real-world situation in which the experimental data has large amounts of random experimental error. Algorithms are optimized and trained on $\text{Train}_{\text{noise}}$, then the resulting model predicts both $\text{Test}_{\text{noise}}$ and $\text{Test}_{\text{true}}$ sets. The quantities $\text{RMSE}_{\text{noise}}$ and $\text{R}^2{}_{\text{noise}}$ are the metrics of the predicted values versus $\text{Test}_{\text{noise}}$ data, replicating the real-world situation where test/validation sets have the same level of noise as the training data. The quantities $RMSE_{true}$ and R^2_{true} are the metrics of the predicted values versus $Test_{true}$ data, our presumed “true” endpoint values. Therefore, the $RMSE_{noise}$ reports on the ability of the algorithm, which is trained on $Train_{noise}$, to predict the noisy values in $Test_{noise}$. In contrast, the $RMSE_{true}$ reports on the ability of the algorithm to predict the values in $Test_{true}$, which have no added noise and thus represent what we can define as “true” values. In this experimental design, for a given noise level, if $RMSE_{true}$ is lower than $RMSE_{noise}$, then the model has made *fewer* errors when predicting the true values.

The results for a representative datasets and algorithm are shown in Figs. 3 and 4 (additional figures for other datasets are available in the Additional file 1). In order to compare trends in data across algorithm and dataset, we chose to normalize the RMSE and the amount of added noise by $RMSE_0$, which is the RMSE obtained from training and predicting on the original noiseless dataset. In the top subplot, the *y*-axis is $RMSE/RMSE_0$. The *x*-axis is the standard deviation of the Gaussian distribution from which the added error was sampled (σ), divided by $RMSE_0$. Therefore, the y-axis indicates the multiplicative increase in observed prediction error, while the x-axis is most accurately understood as the fractional amount of error inserted into the dataset standardized by the amount of prediction error seen in the noiseless dataset. In this figure, a constant value of 1 on the y-axis would corresponds to seeing the exact same error at a particular noise level as was seen when modeling the native dataset. Similarly, a line through the origin with a slope of 1 would represent the expected RMSE obtained if one compared $Test_{noise}$ to $Test_{true}$.

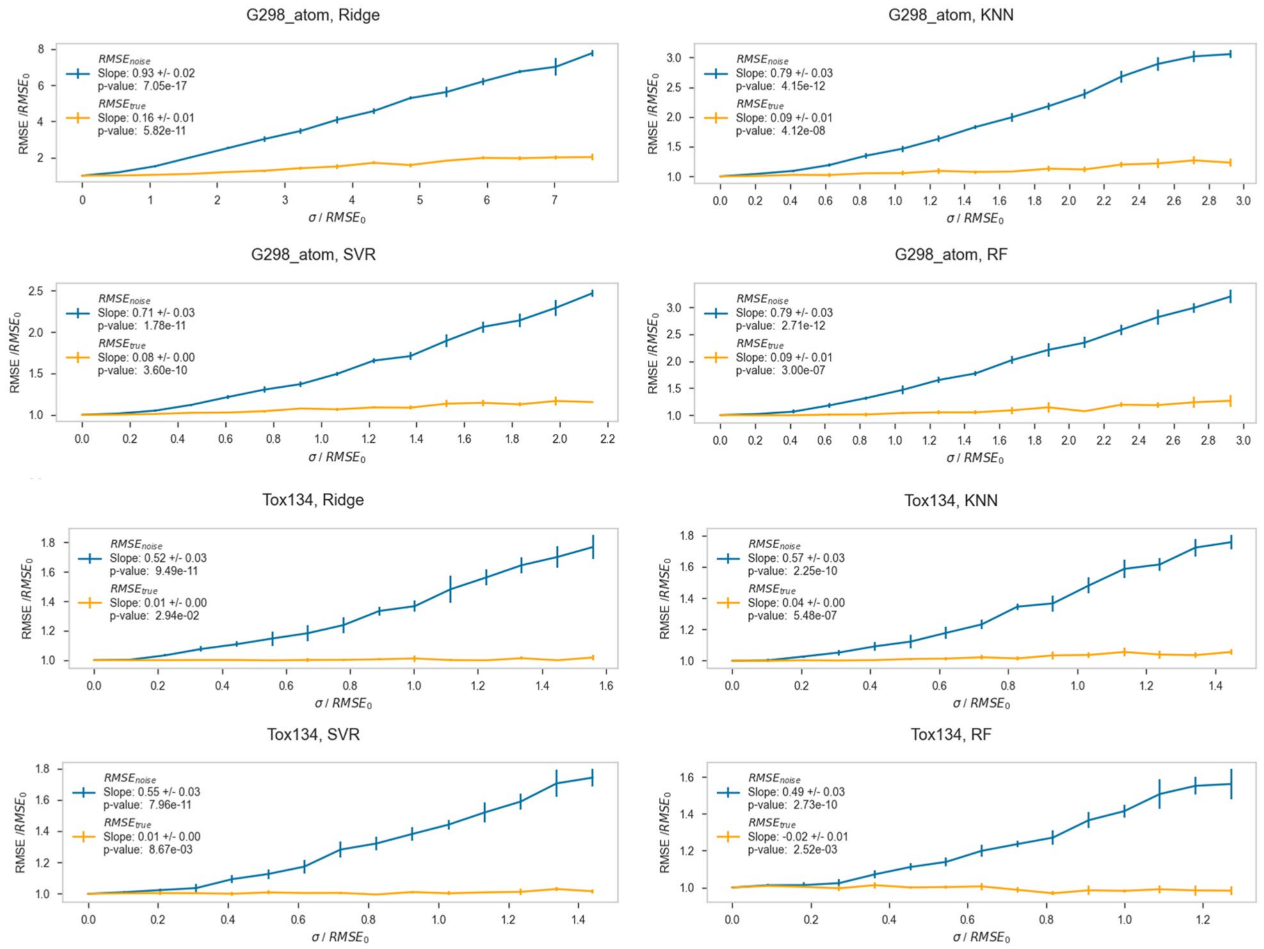


Fig. 3 Plots showing RMSE versus the amount of random error added for the g298_atom and Tox134 datasets, for the Ridge regression, k-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Random Forest (RF) algorithms. For each plot, the *y-axis* is RMSE, and the x-axis is the standard deviation (σ) of the Gaussian distribution of the added error. The blue lines show data evaluated on test sets with error, and orange lines show data evaluated on test sets without error

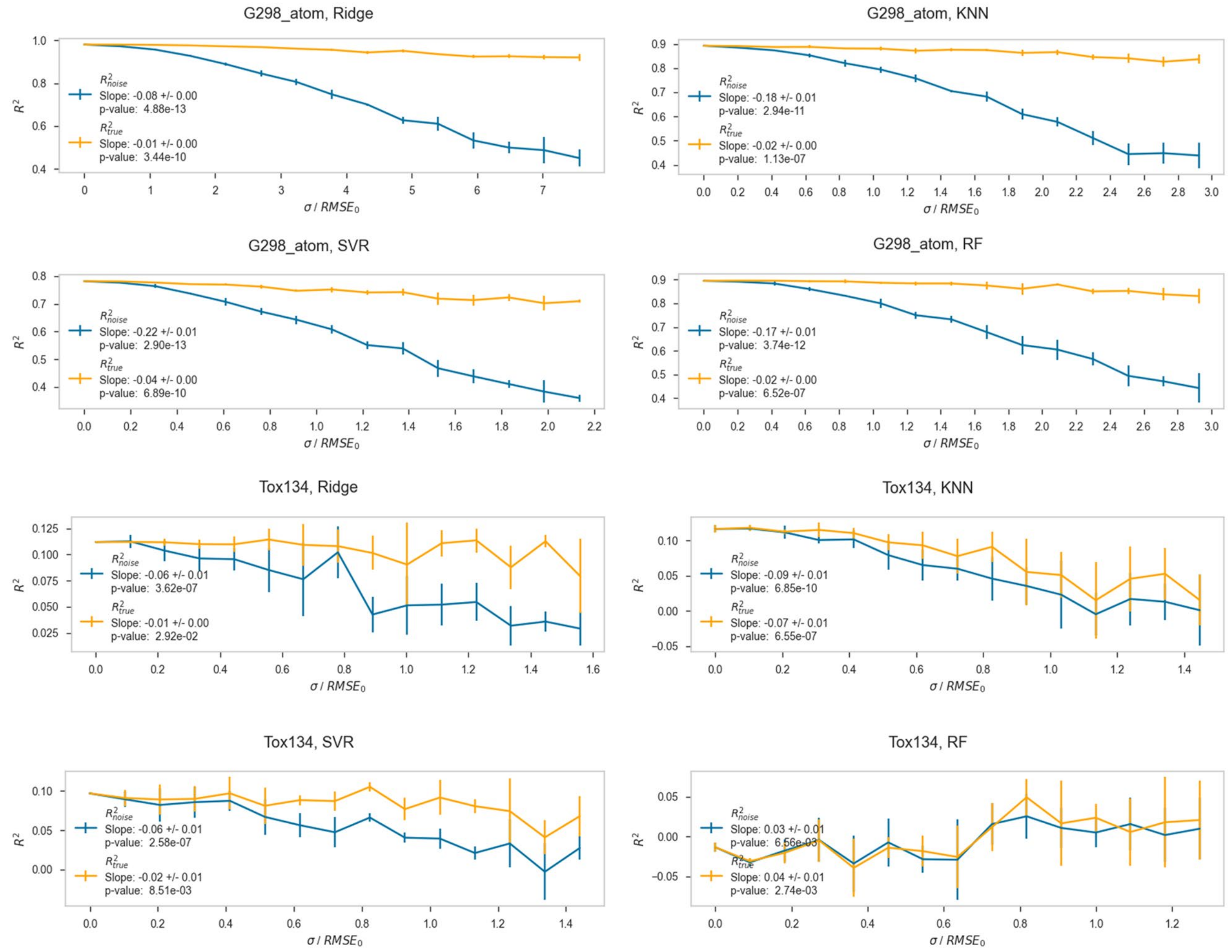


Fig. 4 Plots showing R^2 versus the amount of random error added for the g298_atom and Tox134 datasets, for the Ridge regression, k-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Random Forest (RF) algorithms. For each plot, the *y*-axis is R^2, and the *x*-axis is the standard deviation (σ) of the Gaussian distribution of the added error. The blue lines show data evaluated on test sets with error, and orange lines show data evaluated on test sets without error

For each dataset and algorithm, the $RMSE_{noise}/RMSE_0$ clearly increases as $\sigma/RMSE_0$ increases. The $RMSE_{true}/RMSE_0$ values increase slightly or stay essentially constant, depending on the dataset. What is qualitatively clear from these plots is that $RMSE_{true}/RMSE_0$ stays low and constant, while $RMSE_{noise}/RMSE_0$ rapidly outpaces it. These results, which investigate a variety of different datasets and endpoints, are consistent with the work of Cortés-Ciriano and coworkers, in which, for pIC_{50} datasets, the RMSE on the test set remained constant while noise was added to the training set [22]. The fact that $RMSE_{true}/RMSE_0$ remains nearly constant indicates these models are still accurately predicting the noiseless $Test_{true}$ values despite being trained on increasingly noisy data in $Train_{noise}$. The $RMSE_{noise}$ being consistently higher than $RMSE_{true}$ for each algorithm and dataset indicates that while the models are retaining their accuracy, our ability to validate the models as being accurate using $Test_{noise}$ is significantly degraded. It is also clear that R^2_{noise} and R^2_{true} generally get worse with increasing $\sigma/RMSE_{0,}$ and that R^2_{true} is better than R^2_{noise} for all noise levels. This trend is more apparent in dataset/algorithm combinations which have acceptably large R^2 values, such as the quantum mechanical dataset G298_atom, than in datasets which have extremely low starting R^2 values, such as the toxicological dataset Tox134. Especially with, for example, the combination of Tox134 and RF, both R^2_{noise} and R^2_{true} are 0, indicating that these predictions are not reliable. Having such a small R^2 with a small dynamic range makes forming conclusions about this particular data tenuous at best.

To facilitate the comparison of these trends between algorithms and datasets, a representative quantitative measure of the observed behavior was chosen. First, the slope of $RMSE_{noise}/RMSE_0$ versus $\sigma/RMSE_0$ (m_{noise}) and the slope of $RMSE_{true}/RMSE_0$ versus $\sigma/RMSE_0$ (m_{true}) were obtained by fitting lines to the respective data. These slopes directly report on how $RMSE_{noise}$ and $RMSE_{true}$ behave with the addition of error to the dataset. For example, if m_{noise} is high, then $RMSE_{noise}$ increases significantly as noise is added to the training set, meaning the algorithm becomes worse at predicting $Test_{noise}$ as $Train_{noise}$ becomes noisier. If m_{true} is high, then $RMSE_{true}$ increases significantly as noise is added to the training set, meaning the algorithm is getting worse at predicting $Test_{true}$ as noise is added. The ratio of m_{noise}/m_{true} provides a single metric defining whether a model is predicting closer to true values or noisy values as the training set becomes noiser. If m_{noise}/m_{true} is large, then $RMSE_{noise}$ is increasing much faster than $RMSE_{true}$, and the resultant models are predicting true values much more accurately than noisy values (as they should). This indicates our predictive power on noisy datasets using such an algorithm is likely much better than often perceived from our test/validation statistics. If m_{noise}/m_{true} is close to 1, then $RMSE_{noise}$ and $RMSE_{true}$ are responding very similarly to noise, and the model is not predicting true values much better than noisy values. Table 3 shows m_{noise} and m_{true}, and Table 4 shows m_{noise}/m_{true} ratios.

Table 4 Ratios of m_{noise}/m_{true} for each dataset and algorithm

Dataset/algorithm	Ridge	kNN	SVR	RF
G_298_atom	11 ± 1.3	8.8 ± 1.4	8.9 ± 0.40	8.8 ± 1.6
Alpha	6.6 ± 1.1	8.3 ± 1.3	7.3 ± 1.0	8.1 ± 1.3
Lip	20 ± 12	17 ± 9.0	7.1 ± 1.9	13 ± 5.7
Solv	5.8 ± 1.2	3.0 ± 0.23	3.3 ± 0.26	6.0 ± 0.84
BACE	13 ± 7.7	10 ± 2.8	2.9 ± 0.44	11 ± 3.0
Tox_102	44 ± 3.1	9.2 ± 2.7	–[a]	43 ± 3.0
Tox_134	52* ± 3.1[b]	14 ± 0.84	55 ± 3.3	–[a]
LD_{50}	–[a]	9.7 ± 4.4	5.8 ± 1.2	15 ± 6.3

[a] Entries marked with a—had a null or negative denominator

[b] Entries marked with an * are not statistically significant

Inspecting the values of m_{noise} and m_{true} in Table 3 reveals some consistent behaviors. For a given dataset, m_{noise} and m_{true} are reasonably constant across algorithms (across rows). This observation is consistent with the consistent behavior across algorithms that Cortés-Ciriano observed [22]. For a given algorithm, m (and to a lesser extent m_{true}) vary more significantly over datasets (down columns). This indicates that the RMSE response to added error is consistent for a given dataset with different algorithms, and that the RMSE response is highly variable for a given algorithm across different datasets. These datasets were chosen specifically to encompass a range of experimental complexity and thus a range of native random experimental error. While not definite, the variable nature of the RMSE response to noise over datasets may indicate that these algorithms respond differently to different amounts of native error; Cortés-Ciriano

Table 3 Slopes m_{noise} and m_{true} for each dataset and algorithm

Dataset	Slope	Ridge	kNN	SVR	RF
G298_atom	m_{noise}	0.98 ± 0.011	0.79 ± 0.032	0.71 ± 0.032	0.79 ± 0.030
	m_{true}	0.090 ± 0.010	0.09 ± 0.011	0.08 ± 0.00	0.09 ± 0.013
Alpha	m_{noise}	0.79 ± 0.033	0.83 ± 0.037	0.87 ± 0.023	0.89 ± 0.032
	m_{true}	0.12 ± 0.016	0.10 ± 0.012	0.12 ± 0.014	0.11 ± 0.013
Lip	m_{noise}	0.40 ± 0.031	0.36 ± 0.024	0.44 ± 0.024	0.41 ± 0.031
	m_{true}	0.020 ± 0.011	0.021 ± 0.010	0.062 ± 0.013	0.032 ± 0.012
Solv	m_{noise}	0.75 ± 0.031	0.81 ± 0.031	0.89 ± 0.033	0.72 ± 0.031
	m_{true}	0.13 ± 0.022	0.27 ± 0.011	0.27 ± 0.012	0.12 ± 0.012
BACE	m_{noise}	0.52 ± 0.042	0.53 ± 0.041	0.67 ± 0.033	0.54 ± 0.031
	m_{true}	0.041 ± 0.021	0.052 ± 0.011	0.23 ± 0.023	0.050 ± 0.011
Tox_102	m_{noise}	0.44 ± 0.031	0.49 ± 0.043	0.44 ± 0.031	0.43 ± 0.031
	m_{true}	0.010 ± 0.00	0.053 ± 0.011	0.00*[a]	0.010 ± 0.00
Tox_134	m_{noise}	0.52 ± 0.034	0.57 ± 0.034	0.55 ± 0.031	0.49 ± 0.033
	m_{true}	0.01*[a]	0.041 ± 0.00	0.010 ± 0.00	− 0.020 ± 0.010
LD_{50}	m_{noise}	0.44 ± 0.042	0.43 ± 0.042	0.48 ± 0.033	0.48 ± 0.031
	m_{true}	0.00 ± 0.010	0.044 ± 0.016	0.083 ± 0.012	0.033 ± 0.012

[a] Entries marked with * have *p*-values above 0.05 and thus are not statistically significant

and coworkers observed and commented on the differential response of algorithms to noise, but did not emphasize how noise response differed over different types of endpoints [22]. For example, the quantum mechanical datasets have high m_{noise} values (approaching 1) while toxicity datasets have more moderate slopes (near 0.5). It is expected that higher native error existed in the toxicological datasets compared to the quantum mechanical datasets and such error could have an impact in observing the effects of the additional simulated noise. This suggests that the RMSE response to additional noise likely decreases as the amount of native error in a dataset increases. In contrast, m_{true} varies little and does not follow a decreasing trend over datasets. This observation indicates that these algorithms are capable of finding the "true" values as simulated error was added, regardless of the amount of native error in the original dataset.

Analyzing the m_{noise}/m_{true} ratios in Table 4 reveals how the relative noise responses of $\text{RMSE}_{\text{noise}}$ and $\text{RMSE}_{\text{true}}$ change across algorithm and dataset. One immediate observation is that the ratios for the Tox102 and Tox134 datasets are more highly variable than the ratios of the other datasets. This variability comes from the fact that m_{true} is generally very small, so small changes in this small number lead to large fluctuations in the m_{noise}/m_{true} ratios. This instability could be viewed as one detriment of this metric. It is also apparent that the Tox102 and Tox134 datasets have the highest ratios, albeit with large variability. This means that as noise is added to these datasets, $\text{RMSE}_{\text{noise}}$ increases much more rapidly than $\text{RMSE}_{\text{true}}$, and the algorithms can predict the true values more accurately than the noisy values. We expect that Tox102 and Tox134 have relatively high native error compared to the quantum mechanical, physiochemical, and biochemical datasets, and we propose that the addition of more error to these datasets does not affect the algorithms ability to predict the true values as drastically as it does to the other datasets. This proposal is supported by the fact that the m_{true} values for Tox102 and Tox134 are roughly an order of magnitude smaller than m_{true} values for the G298_atom, Alpha, and Solv datasets.

These experiments used PCA to reduce the number of descriptors involved in prediction while maintaining as much variance as possible. Using PCA achieves dimension reduction by forming linear combinations of the original descriptors; although this process ultimately reduces the physical interpretability of the model, it does provide a significant computational advantage because the predictive algorithm has fewer, but more information dense, variables to work with. However, given that this preprocessing step is somewhat uncommon in the QSAR literature, the effect of using PCA on the m_{noise}/m_{true} ratios was tested. The results without PCA in Table 5 show mixed trends when compared with Table 4. The most dramatic effect is seen across each dataset using the Ridge algorithm, for which the ratios all drop significantly. This is expected because Ridge regression uses a regularization to mitigate variance at the expense of adding bias; this means that the algorithm is sensitive to having many feature variables that complicate finding a useful trend. Therefore, when PCA is not used, the Ridge algorithm does not predict the true values as well and the ratio decreases. For kNN and SVR however, the ratios are not sensitive to the use of PCA. This experiment was not carried out for RF because computational time scales with the number of descriptors, so performing the workflow without dimension reduction made the calculation time unreasonable. The other apparent trend is that the ratios for the Tox102 and Tox134 datasets are significantly reduced without PCA. This result suggests that predicting the true values in these datasets is sensitive to the number of descriptors, so that when many extraneous descriptors are used the ratios become smaller.

Table 5 Ratios of m_{noise}/m_{true} without Principal Component Analysis

Dataset/algorithm	Ridge	kNN	SVR
G_298_atom	1.4 ± 0.10	8.0 ± 1.4	5.1 ± 0.22
Alpha	1.7 ± 0.13	13 ± 3.4	4.7 ± 0.58
Lip	1.9 ± 0.53	12 ± 5.0	3.1 ± 0.26
Solv	1.4 ± 0.080	2.5 ± 0.24	3.3 ± 0.48
BACE	1.6 ± 0.10	6.8 ± 0.95	14 ± 0.64
Tox_102	1.5 ± 0.075	6.4 ± 0.70	7.6 ± 0.29
Tox_134	1.0 ± 0.15	7.8 ± 1.2	10 ± 0.44
LD_{50}	1.3 ± 0.15	7.5 ± 1.0	31 ± 1.7

Additionally, it is useful to contextualize the amount of simulated error which has been added to these datasets within what is known about experimental uncertainties. Estimates for most of the endpoints used in this study are not readily available, however Kramer, Kalliokoski and colleagues found from an examination of the ChemBL database that heterogeneous pIC_{50} data has an average standard deviation of 0.68 log units [19]. For the BACE dataset, which uses a pIC_{50} endpoint, 1.1 log units of noise were added, or 1.6 times the average standard deviation reported in ChemBL.

Gaussian process results

In addition to quantifying how *accurate* QSAR predictions are, it is very useful to quantify how *precise* predictions are. While machine learning algorithms such as Ridge regression, kNN, SVR, and RF do not have a direct method of quantifying the precision or uncertainty of the predictions made on each molecule, the Gaussian Process (GP) algorithm does provide direct uncertainties for each of its predictions. We utilized the GP algorithm to investigate how prediction precision is affected by the addition of simulated error into each dataset.

There has been extensive work carried out on the general topic of prediction uncertainties in the QSAR literature, typically involving Bayesian methods. Wood and coworkers analyzed model output with the Kullback–Leibler divergence to generate estimates of prediction uncertainty [55]. Burden introduced GP to the QSAR community [56], Obrezanova and coworkers later applied GP to ADME properties, highlighting its usefulness as an application in drug discovery [57, 58], and many other works have utilized GP with other endpoints [59–61]. Cortés-Ciriano and coworkers applied GP to the field of proteochemometrics, using the prediction uncertainty to estimate the applicability domain of the model [62]. Conformal prediction is a non-Bayesian technique which also produces confidence intervals, and has been applied often in QSAR and computational toxicology [63–71]. Conformal prediction has the advantage that it does not require the selection of a prior distribution like GP, which means that no assumptions need to be made about the underlying distribution of the data. While a quantitative comparison of conformal prediction and GP is outside the scope of this work, the comparison has been made elsewhere [72]. The advantage of providing prediction uncertainties in the field of QSAR motivated the study of GP in this work, in order to understand how the addition of noise to the various datasets affects the precision of the predictions.

Following the analysis of the Results section for the other algorithms, we examined RMSE and R^2 for GP to give a measure of prediction accuracy. However, to quantify prediction precision, we examined the prediction uncertainty $\sigma_{\hat{y}}$, or width of each individual prediction. We examined both the mean $\sigma_{\hat{y}}$ (Eq. 21), which is the average of all the individual prediction uncertainties, and the $\sigma_{\hat{y}}$ 95% confidence interval (Eq. 22), which is the spread of the individual prediction uncertainties. An important point of emphasis is that the prediction uncertainty $\sigma_{\hat{y}}$ is completely dependent on the descriptor values and is independent of whether the prediction is evaluated using the true test set or the noisy test set. In other words, the precision of a prediction is completely dependent on how close that molecule is in feature space to other molecules. This behavior contrasts with the metrics RMSE and R^2, which depend entirely on whether the "true" answer comes from a true test set or a noisy test set.

The GP algorithm also has the option to include information about the uncertainty of the experimental measurement vector *Y;* we will define this uncertainty vector as σ_y (Eq. 24). When σ_y is given to GP, the algorithm can incorporate this information to adjust the precision of each element in its prediction vector $\hat{Y}$. Ignoring native error in the datasets, the uncertainty in the measurements σ_y is just the width of the gaussian distribution from which the error was sampled; each term σ_{yn} within the vector σ_y will be the same.

$$\hat{Y} = \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n \tag{19}$$

$$\sigma_{\hat{y}} = \sigma_{\hat{y}1}, \sigma_{\hat{y}2}, \ldots, \sigma_{\hat{y}n} \tag{20}$$

$$Mean\sigma_{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n}\sigma_i \tag{21}$$

$$\sigma_{\hat{y}}95\%CI = \frac{1.960}{\sqrt{n}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\sigma_i - Mean\sigma_{\hat{y}}\right)^2\right] \tag{22}$$

$$Y = y_1, y_2, \ldots, y_n \tag{23}$$

$$\sigma_y = \sigma_{y1}, \sigma_{y2}, \ldots, \sigma_{yn} \tag{24}$$

Plots of a selection of GP results are shown in Fig. 5, and the tabulated GP prediction accuracy results are shown in Table 6. Each row gives the ratio of m_{noise} to m_{true} for each dataset. The first column shows values for which uncertainty in the *Y* vector was not provided, and the second column shows values for which uncertainty in the *Y* vector was provided. For the first column, without σ_y, the m_{noise}/m_{true} ratios are all greater than 1, however they are somewhat lower than average than the ratios for the other algorithms, which are shown in Table 4. This result suggests that GP when instructed to assume no uncertainty exists in its training set is not as robust to error inserted into its training set, at least in comparison to the other algorithms in this study. However, in the second column, where σ_y was provided to GP, the ratios increased. The m_{noise}/m_{true} ratio approximately doubled for the BACE dataset, and more than quadrupled for the Alpha dataset. This result shows that when information about the uncertainty in the measurements is known, the GP algorithm makes predictions which are much closer to the true values than the artificially noisy values with a similar robustness to other learning algorithms. The ability to directly incorporate measurement uncertainty into predictions is a unique feature of the GP algorithm and provides useful insight into how prediction accuracy and precision are connected to experimental error.

As mentioned above, GP provides quantitative information about the uncertainty in its predictions, which is contained in the vector $\sigma_{\hat{y}}$. In order to investigate how

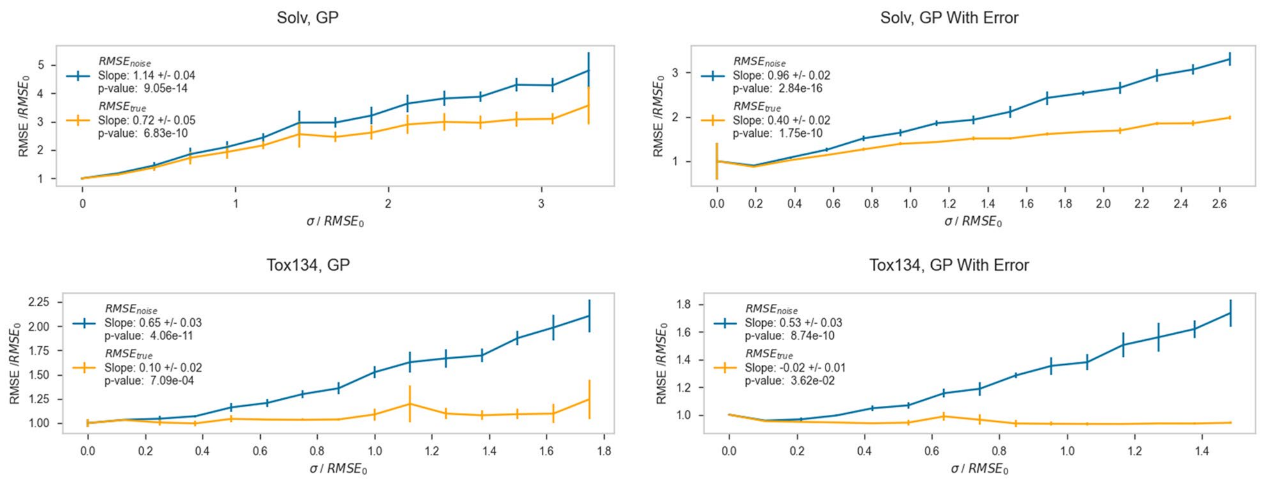


Fig. 5 Plots showing RMSE versus the amount of added error to the Solv and Tox134 datasets, using Gaussian Process algorithm. Plots on the left are for data where no information about the added error *has not* been fed to the GP algorithm, and plots on the right are for data where added error information *has* been fed to the GP algorithm

Table 6 Ratios of *m* to m_{true} for the Gaussian Process algorithm

Dataset	No σ_y	With σ_y
G_298_atom	6.9 ± 1.5	1.7 ± 0.26
Alpha	1.8 ± 0.11	9.3 ± 0.37[a]
Solv	1.6 ± 0.24	2.4 ± 0.17[a]
BACE	3.7 ± 2.0	8.6 ± 1.6[a]
Tox_102	2.7 ± 1.7	_[b]
Tox_134	6.5 ± 1.6	_[b]
LD_{50}	5.3 ± 0.74	5.5 ± 0.83

[a] Slopes m_{noise} and m_{true} were calculated excluding the first point due to a discontinuity in the line

[b] The slope m_{true} was negative for these plots, so the slope ratio was not calculated

Table 7 Slopes of mean $\sigma_{\hat{y}}$ and $\sigma_{\hat{y}}$ 95% CI versus σ for the Gaussian Process algorithm. Results are shown with and without the input of σ_y into the algorithm

Dataset	No σ_y	No σ_y	With σ_y	With σ_y
	Mean σ_y	σ_y 95% CI	Mean σ_y	σ_y 95% CI
G_298_atom	1.0	0.40	0.52	− 0.10
Alpha	1.1	0.16	0.44[a]	0.32[a]
Solv	0.94	− 0.19	0.10	0.10
BACE	0.25	0.38	− 0.12	− 0.35
Tox_102	0.32	0.028	− 0.96	− 0.48
Tox_134	0.49	0.53	− 0.66	− 0.17
LD_{50}	0.66	− 0.39	− 0.60	0.14

[a] The first point was omitted in these calculations because of a discontinuity in the line

prediction precision is affected by the addition of simulated error, the prediction uncertainty $\sigma_{\hat{y}}$ was plotted versus the amount of added error σ. Additionally, the effect of providing measurement uncertainty σ_y to the GP algorithm was explored. Table 7 shows the slopes of mean $\sigma_{\hat{y}}$ versus σ and the $\sigma_{\hat{y}}$ 95% confidence interval versus σ, for GP models where σ_y was and was not provided. Slopes were obtained by linear fits to the data, although in some cases the data was significantly non-linear; while this analysis is imperfect, we feel that it still allows useful qualitative trends to be captured.

Inspection of Table 7 and the accompanying Fig. 6 show that when the measurement uncertainty σ_y is withheld from the algorithm, the slopes of mean $\sigma_{\hat{y}}$ versus σ are all positive. This indicates that prediction precision gets worse as noise is added into the data. These slopes also generally become smaller as the qualitative complexity of the datasets increase. This could be attributed to the amount of native error present in each dataset. For example, while the G298atom dataset has no experimental uncertainty because it is composed of quantum mechanical endpoints, the Tox102 dataset is composed of in vitro measurements with a large degree of variability. Because the Tox102 dataset contains more native error, the prediction precision is not as sensitive to the addition of noise.

The slope of prediction uncertainty $\sigma_{\hat{y}}$ is very sensitive to the inclusion of measurement uncertainty σ_y. Including measurement uncertainty in the calculation decreases the slope for each of the datasets, even causing some of the slopes to become negative. This indicates that information about the variability in the

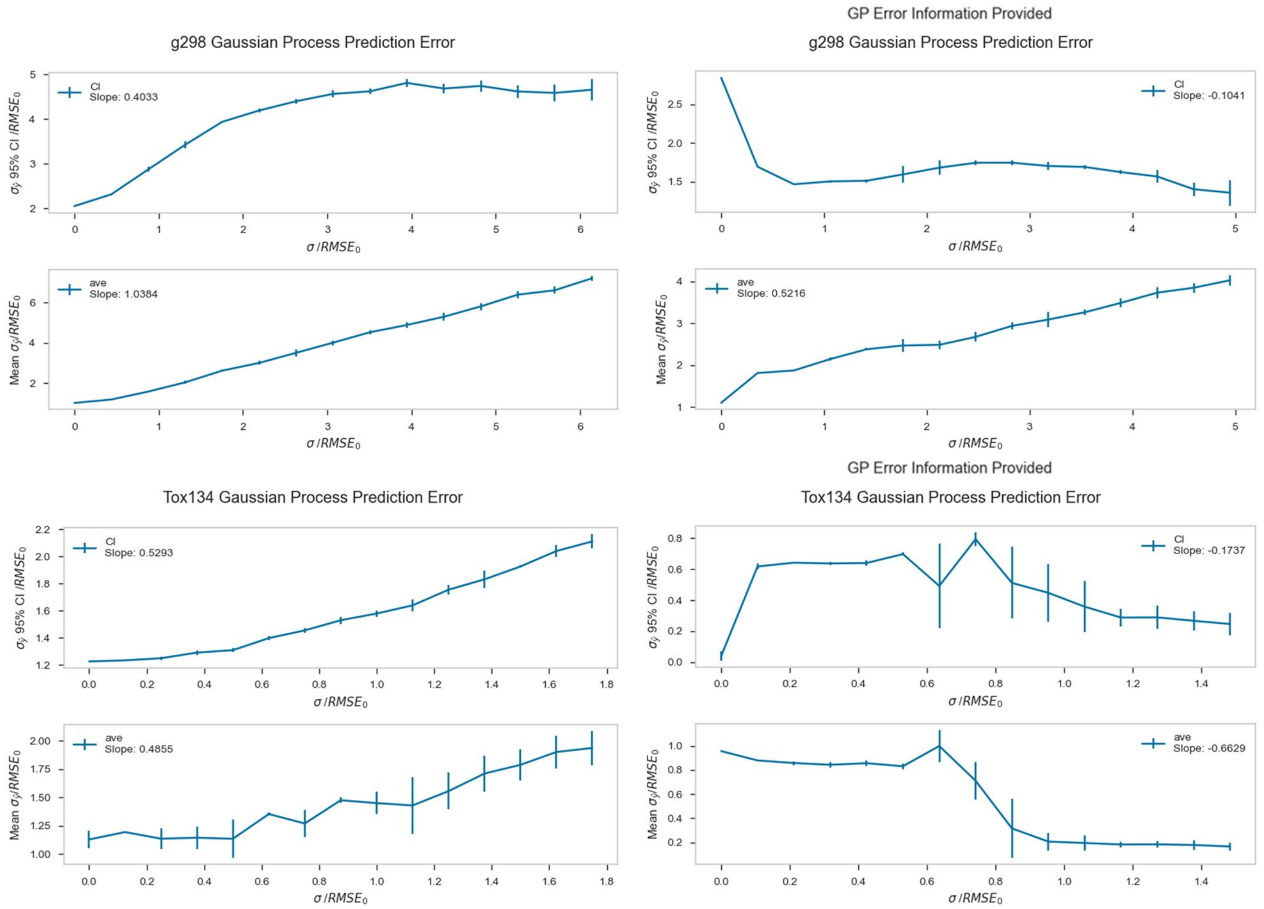


Fig. 6 Plots showing prediction error versus amount of added error to the G298_atom and Tox134 datasets, using the Gaussian Process algorithm. The top subplots show the average 95% confidence interval for prediction error versus added error. The bottom subplots show the mean prediction error versus amount of added error

measurements reduces the effect that added error has on the prediction precision. This reductive effect is mild for the quantum mechanical and physiochemical datasets but becomes more pronounced for the in vitro and in vivo datasets. This result shows that even when datasets have large uncertainty in the measurements, the predictions from GP can apparently become *more precise* as more error is introduced as long as the magnitude of that error is known, the error is normally distributed, and the error is provided as an input. Error in datasets is not always known, nor is it always normally distributed. The experiments described here nevertheless provide a foundation for understanding how the effect of added error can be mitigated when using Gaussian Processes, when the nature of that error is known. Because of the nature of this experiment, the distribution and magnitude of the error was predetermined, which, admittedly, is not a situation that is common in QSAR modeling.

The 95% confidence interval of $\sigma_{\hat{y}}$ shows more complicated behavior as error is added to the datasets. When measurement uncertainty is withheld from the algorithm, the slope of 95% confidence interval versus σ is positive for each dataset except Solv and LD_{50}, which show negative trends. Additionally, the G298atom and Alpha datasets show a quadratic trend which levels off at high values of σ, which contrasts with the more linear trends observed in the other datasets. This indicates that, generally, the distribution of prediction error is getting larger as more error is added to the datasets. In other words, as more error is added to the datasets, not only does the average prediction uncertainty increase, but the spread in those average uncertainties becomes larger as well. It remains unclear why this behavior is different for the Solv and Tox134 datasets. Although the Solv dataset shows a relatively flat slope, the LD_{50} dataset shows a clearly negative trend.

When measurement uncertainty is provided to the GP algorithm, the trends in the 95% confidence interval of $\sigma_{\hat{y}}$ change. The change in behavior is inconsistent and complicated across the datasets but including information about measurement uncertainty clearly affects the trends significantly. One consistent effect is that the error bars become much smaller, which shows that the results are much more tightly distributed between the 5 repetitions at each level of σ added to the datasets.

Additionally, it is possible to evaluate the mean prediction uncertainty that GP provides by comparing it to the mean experimental estimate of uncertainty for pIC_{50} provided by Kolliokoski and colleagues [19]. The mean $\sigma_{\hat{y}}$ can be obtained using $RMSE_0$ and σ_{noise} of 0 for the GP calculations on the BACE dataset. Using the $RMSE_0$ value of 0.98 for the GP calculations on the BACE dataset, the mean $\sigma_{\hat{y}}$ is 0.79 log units. The estimated experimental uncertainty for pIC50 is 0.68 log units, so GP's prediction uncertainty is 1.2 times the experimental estimate, when no simulated noise has been added to the dataset.

Discussion and conclusions

The purpose of this work is to examine the common assumption that QSAR models cannot make predictions which are more accurate than their training data. Many other works have contributed to this general topic, including thorough estimations of the random error in K_i, [20] IC_{50}, [19] and cytotoxicity [21] databases and an investigation of the noise tolerance of machine learning algorithms with IC_{50} data [22]. These works and others have supported the well-known phenomenon that machine learning algorithms are generally tolerant to noise. There is a general contention however that experimental uncertainty sets the upper limit of in silico predictions [20], and this study has attempted to examine that assertion. This work has attempted to ask, in the presence of increasingly noisy data, if these algorithms can formulate a trend that predicts closer to the true values than the artificial noisy values. However, investigation of this central hypothesis has two main limitations. The first limitation is statistical, which is that experimental values are typically only single values. When multiple values are available, there are still too few to reliably approximate the population mean for the measurement. This means that QSAR models are built on data which may poorly capture the physical reality of the trends being modeled. This limitation is recognized by the field, but there is little that can be done without increasing the rate of experimentation. The second limitation is the assumption that test sets and validation sets have no associated error, or at least this assumption is necessitated by the methods used. Because QSAR models are evaluated on these test and validation sets, this means that QSAR models are being judged by their ability to predict error laden values, when they should be judged by their ability to predict the population means of measurements. The result of these limitations is that it is commonly assumed/stated that QSAR models cannot make predictions which are "better" or more accurate than their training data. A more exact statement would be that cross/external validation statistics (our standard metrics of predictivity) for QSAR models are limited based on the accuracy of the dataset. The present work has designed a set of experiments to examine these limitations and this hypothesis by adding simulated error into a variety of representative QSAR datasets and designating two classes of test sets. The first class of test set comes from "true" error free values, and the second class of test set comes from the "noisy" error laden values. The difference in performance metrics between these two classes of test sets allows us to examine whether models can really generate predictions which are more accurate than the noisy data they were trained on. The error added to the datasets in this work was Gaussian distributed, which provides a convenient analogy for real-world data situations in which endpoint values fall somewhere on a Gaussian distribution of error. It is true that this situation is not always the case. Despite the fact that the present experiments are testing a hypothesis that could be labeled an "ideal" case of dataset error, we posit that it still provides useful conclusions that have not been clearly stated in QSAR modeling literature.

The results show that there is a consistent difference in the RMSE when predictions are evaluated against the true and noisy test sets, across 5 algorithms and 8 datasets. The $RMSE_{true}$ values are all lower than the corresponding RMSE values. When increasing amounts of error were added to the datasets, the difference between $RMSE_{true}$ and RMSE became larger. This indicates that these models are predicting true values more accurately than noisy values, even when the algorithms are trained on data with large amounts of added simulated error. This scenario mirrors what likely happens for many QSAR models. A model is built on data with an unknown amount of error, which means that each experimental value may fall an unknown distance away from the true population mean for that measurement. Evaluation statistics for the QSAR model are then generated on internal test sets or an external validation set which are composed of values with unknown amounts of error. The RMSE, when calculated for these test sets, may be quite high, and thus the model is judged to be flawed. Work examining uncertainty in pK_i data asserts that if the uncertainty in training and validation sets are comparable, then the minimum RMSE obtainable should be equivalent to the uncertainty in the experimental data [20]. While this applies to situations in which experimental uncertainty estimates are available, it does not as readily apply when these estimations are unavailable. These results show that those models may very well be predicting the population means of those measurements, but this fact is obscured by the error in the test sets. Even from a very conservative interpretation of the results shown here, this study indicates that this situation is plausible.

The results also show that the difference between the observed RMSE and the unknown $\text{RMSE}_{\text{true}}$ depends on algorithm and dataset complexity. This is an important observation, because it suggests that when models using different algorithms are compared, they may have significantly different accuracies, even if the observed RMSEs are very close. For example, examining the Solv row in Table 3, the m_{noise}/m_{true} ratio is 3.3 for SVR and 6.1 for RF. This means that in a real modeling situation, if these SVR and RF algorithms produced the same RMSE for the Solv dataset, the $\text{RMSE}_{\text{true}}$'s (and the relevant comparison) would be different by a factor of 1.8. Because real world datasets are undeniably rife with unknown amounts of error, this example demonstrates that comparing QSAR models through error laden test sets may be producing misleading conclusions in terms of model performance.

It is important to recognize that error in training sets appears to result in only a minor increase in "true" predictive error as assumed in this work (at least when work with datasets containing 1000 datapoints). In general, QSAR evaluation techniques cause us to perceive large amounts of predictive error when our training sets have error; this phenomenon is represented by the large $\text{RMSE}_{\text{noise}}$ (what is observable in the general case) compared to the small $\text{RMSE}_{\text{true}}$ (what unobservable in the general case). These observations were made by Cortés-Ciriano and coworkers on pIC_{50} datasets, and the current work complements and extends those initial studies [22]. Therefore, new learning methods will not resolve the issue. While some methods like Gaussian Processes and Conformal Prediction take error into account as part of training and allow modelers to estimate prediction precision, there are associated limitations. Conformal Prediction requires that a segment of the training set be put aside for calibration, while Gaussian Process requires a reasonable prior distribution and some knowledge of the experimental uncertainty to be effective. Much effort has been given towards analyzing experimental uncertainties for endpoints such as pK_{i}, [20] pIC_{50}, [19] and cytotoxicity [21] using public databases, providing useful inputs for methods like Gaussian Process and Conformal Prediction. Efforts towards estimating uncertainties of other common QSAR endpoints would be welcome.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-021-00571-7.

Additional file 1. Information about datasets and supplemental plots can be found in the additional file.

Acknowledgements

The authors gratefully acknowledge colleagues at the US Environmental Protection Agency, Charles Lowe and Richard Judson, for providing technical reviews of this manuscript.

Authors' contributions

The manuscript was written through contributions of all authors. All authors read and approved the final manuscript.

Funding

The United States Environmental Protection Agency through its Office of Research and Development funded and managed the research described here.

Availability of data and materials

All code used in this work, as well as the datasets containing SMILES strings, QSAR ready SMILES strings, and descriptor values, can be found in our public GitHub repository here: https://github.com/USEPA/CompTox-ChemInf-ModelExperiments-ErrorEffects/tree/SIRepo. All the necessary information to inspect the code and reproduce the results in this work can be found in the repository.

Declarations

Competing interests

The authors declare no competing interests.

Received: 15 July 2021 Accepted: 14 November 2021
Published online: 25 November 2021

References

1. Golbraikh A, Tropsha A (2002) Beware of q2! J Mol Graph Model 20(4):269–276
2. Alexander T, Alexander G (2007) Predictive QSAR modeling workflow, model applicability domains, and virtual screening. Curr Pharm Des 13(34):3494–3504
3. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inf 29(6–7):476–488
4. Cherkasov A, Muratov EN, Fourches D, Varnek A, Baskin II, Cronin M, Dearden J, Gramatica P, Martin YC, Todeschini R, Consonni V, Kuz'min VE, Cramer R, Benigni R, Yang C, Rathman J, Terfloth L, Gasteiger J, Richard A, Tropsha A (2014) QSAR modeling: where have you been? Where are you going to? J Med Chem 57(12):4977–5010
5. Muratov EN, Bajorath J, Sheridan RP, Tetko IV, Filimonov D, Poroikov V, Oprea TI, Baskin II, Varnek A, Roitberg A, Isayev O, Curtalolo S, Fourches D, Cohen Y, Aspuru-Guzik A, Winkler DA, Agrafiotis D, Cherkasov A, Tropsha A (2020) QSAR without borders. Chem Soc Rev 49(11):3525–3564
6. Brown SP, Muchmore SW, Hajduk PJ (2009) Healthy skepticism: assessing realistic model performance. Drug Discov Today 14(7):420–427
7. Wenlock MC, Carlsson LA (2015) How experimental errors influence drug metabolism and pharmacokinetic QSAR/QSPR models. J Chem Inf Model 55(1):125–134
8. Pham LL, Watford SM, Pradeep P, Martin MT, Thomas RS, Judson RS, Setzer RW, Friedman KP (2020) Variability in in vivo studies: defining the upper limit of performance for predictions of systemic effect levels. Comput Toxicol 15:100126
9. Jaworska JS, Comber M, Auer C, Leeuwen CJV (2003) Summary of a workshop on regulatory acceptance of (Q)SARs for human health and environmental endpoints. Environ Health Perspect 111(10):1358–1360

10. OECD principles for the validation, for regulatory purposes, Of (quantitative) structure-activity relationship models. https://www.oecd.org/chemicalsafety/risk-assessment/37849783.pdf. Accessed 19 Nov 2020
11. Miller J, Miller JC (2018) Statistics and chemometrics for analytical chemistry. Pearson education, London
12. Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning, vol 2. MIT press, Cambridge
13. Shafer G, Vovk V (2008) A tutorial on conformal prediction. J Mach Learn Res 9:371–421
14. Vovk V, Gammerman A, Shafer G (2005) Algorithmic learning in a random world. Springer, Berlin
15. Watt ED, Judson RS (2018) Uncertainty quantification in ToxCast high throughput screening. PloS ONE 13(7):e0196963
16. Webb GI (2010) Overfitting. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning. Boston, Springer, pp 744–744
17. Gauss CF(1877) Theoria motus corporum coelestium in sectionibus conicis solem ambientium. FA Perthes, Gothae
18. Le Cam L (1935) The central limit theorem around. Stat Sci 1986:78–91
19. Kalliokoski T, Kramer C, Vulpetti A, Gedeck P (2013) Comparability of mixed IC50 data—a statistical analysis. PloS ONE 8(4):e61007
20. Kramer C, Kalliokoski T, Gedeck P, Vulpetti A (2012) The experimental uncertainty of heterogeneous public Ki data. J Med Chem 55(11):5165–5173
21. Cortés-Ciriano I, Bender A (2016) How consistent are publicly reported cytotoxicity data? Large-scale statistical analysis of the concordance of public independent cytotoxicity measurements. ChemMedChem 11(1):57–71
22. Cortes-Ciriano I, Bender A, Malliavin TE (2015) Comparing the influence of simulated experimental errors on 12 machine learning algorithms in bioactivity modeling using 12 diverse data sets. J Chem Inf Model 55(7):1413–1425
23. Casati S, Aschberger K, Barroso J, Casey W, Delgado I, Kim TS, Kleinstreuer N, Kojima H, Lee JK, Lowit A, Park HK, Régimbald-Krnel MJ, Strickland J, Whelan M, Yang Y, Zuang V (2018) Standardisation of defined approaches for skin sensitisation testing to support regulatory use and international adoption: position of the international cooperation on alternative test methods. Arch Toxicol 92(2):611–617
24. Thomas RS, Bahadori T, Buckley TJ, Cowden J, Deisenroth C, Dionisio KL, Frithsen JB, Grulke CM, Gwinn MR, Harrill JA, Higuchi M, Houck KA, Hughes MF, Hunter ES III, Isaacs KK, Judson RS, Knudsen TB, Lambert JC, Linnenbrink M, Martin TM, Newton SR, Padilla S, Patlewicz G, Paul-Friedman K, Phillips KA, Richard AM, Sams R, Shafer TJ, Setzer RW, Shah I, Simmons JE, Simmons SO, Singh A, Sobus JR, Strynar M, Swank A, Tornero-Valez R, Ulrich EM, Villeneuve DL, Wambaugh JF, Wetmore BA, Williams AJ (2019) The next generation blueprint of computational toxicology at the US environmental protection agency. Toxicol Sci 169(2):317–332
25. Claassen V (2013) Neglected factors in pharmacology and neuroscience research: biopharmaceutics, animal characteristics, maintenance, testing conditions, vol 12. Elsevier, Amsterdam
26. Truong L, Ouedraogo G, Pham L, Clouzeau J, Loisel-Joubert S, Blanchet D, Noçairi H, Setzer W, Judson R, Grulke C, Mansouri K, Martin M (2018) Predicting in vivo effect levels for repeat-dose systemic toxicity using chemical, biological, kinetic and study covariates. Arch Toxicol 92(2):587–600
27. Mazzatorta P, Estevez MD, Coulet M, Schilter B (2008) Modeling oral rat chronic toxicity. J Chem Inf Model 48(10):1949–1954
28. Lejaeghere K, Van Speybroeck V, Van Oost G, Cottenier S (2014) Error estimates for solid-state density-functional theory predictions: an overview by means of the ground-state elemental crystals. Crit Rev Solid State Mater Sci 39(1):1–24
29. Sim E, Song S, Burke K (2018) Quantifying density errors in DFT. J Phys Chem Lett 9(22):6385–6392
30. Abraham MH, Whiting GS, Fuchs R, Chambers EJ (1990) Thermodynamics of solute transfer from water to hexadecane. J Chem Soc Perkin Trans 2. https://doi.org/10.1039/P29900000291
31. Poole CF (2004) Chromatographic and spectroscopic methods for the determination of solvent properties of room temperature ionic liquids. J Chromatogr A 1037(1):49–82
32. Jarmoskaite I, AlSadhan I, Vaidyanathan PP, Herschlag D (2020) How to measure and evaluate binding affinities. Life 9:e57264
33. Judson RS, Magpantay FM, Chickarmane V, Haskell C, Tania N, Taylor J, Xia M, Huang R, Rotroff DM, Filer DL, Houck KA, Martin MT, Sipes N, Richard AM, Mansouri K, Setzer RW, Knudsen TB, Crofton KM, Thomas RS (2015) Integrated model of chemical perturbations of a biological pathway using 18 in vitro high-throughput screening assays for the estrogen receptor. Toxicol Sci 148(1):137–154
34. Richard AM, Judson RS, Houck KA, Grulke CM, Volarath P, Thillainadarajah I, Yang C, Rathman J, Martin MT, Wambaugh JF, Knudsen TB, Kancherla J, Mansouri K, Patlewicz G, Williams AJ, Little SB, Crofton KM, Thomas RS (2016) ToxCast chemical landscape: paving the road to 21st century toxicology. Chem Res Toxicol 29(8):1225–1251
35. Blum LC, Reymond J-L (2009) 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. J Am Chem Soc 131(25):8732–8733
36. Ramakrishnan R, Dral PO, Rupp M, von Lilienfeld OA (2014) Quantum chemistry structures and properties of 134 kilo molecules. Sci Data 1(1):140022
37. Wenlock M, Tomkinson N. ChEMBL. https://www.ebi.ac.uk/chembl/document_report_card/CHEMBL3301361/
38. Mobley DL, Guthrie JP (2014) FreeSolv: a database of experimental and calculated hydration free energies, with input files. J Comput Aided Mol Des 28(7):711–720
39. Subramanian G, Ramsundar B, Pande V, Denny RA (2016) Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches. J Chem Inf Model 56(10):1936–1949
40. Wu Z, Ramsundar B, Feinberg Evan N, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) MoleculeNet: a benchmark for molecular machine learning. Chem Sci 9(2):513–530
41. Gadaleta D, Vuković K, Toma C, Lavado GJ, Karmaus AL, Mansouri K, Kleinstreuer NC, Benfenati E, Roncaglioni A (2019) SAR and QSAR modeling of a large collection of LD50 rat acute oral toxicity data. J Cheminform 11(1):58
42. PadelPy GitHub. https://github.com/ECRL/PaDELPy. Accessed 3 Jan 2021
43. Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474
44. Padel Software URL. http://www.yapcwsoft.com/dd/padeldescriptor/. Accessed 3 Jan 2021
45. Mansouri K, Grulke CM, Judson RS, Williams AJ (2018) OPERA models for predicting physicochemical properties and environmental fate endpoints. Journal of Cheminformatics 10(1):10
46. OPERA Github. https://github.com/kmansouri/OPERA
47. Sagarika S, Chandana A, Minati K, Bijay KM (2016) A short review of the generation of molecular descriptors and their applications in quantitative structure property/activity relationships. Curr Comput Aided Drug Des 12(3):181–205
48. Karelson M, Lobanov VS, Katritzky AR (1996) Quantum-chemical descriptors in QSAR/QSPR studies. Chem Rev 96(3):1027–1044
49. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inform Comput Sci 28(1):31–36
50. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
51. Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67
52. Silverman BW, Jones MC (1989) E. Fix and J.L. Hodges (1951): an important contribution to nonparametric discriminant analysis and density estimation: commentary on Fix and Hodges (1951). Int Stat Rev 57(3):233–238
53. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
54. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
55. Wood DJ, Carlsson L, Eklund M, Norinder U, Stålring J (2013) QSAR with experimental and predictive distributions: an information theoretic approach for assessing model quality. J Comput Aided Mol Des 27(3):203–219
56. Burden FR (2001) Quantitative structure—activity relationship studies using gaussian processes. J Chem Inf Comput Sci 41(3):830–835
57. Obrezanova O, Csányi G, Gola JMR, Segall MD (2007) Gaussian processes: a method for automatic QSAR modeling of ADME properties. J Chem Inf Model 47(5):1847–1857

58. Obrezanova O, Segall MD (2010) Gaussian processes for classification: QSAR modeling of ADMET and target activity. J Chem Inf Model 50(6):1053–1061
59. Schwaighofer A, Schroeter T, Mika S, Laub J, ter Laak A, Sülzle D, Ganzer U, Heinrich N, Müller K-R (2007) Accurate solubility prediction with error bars for electrolytes: a machine learning approach. J Chem Inf Model 47(2):407–424
60. Romero PA, Krause A, Arnold FH (2013) Navigating the protein fitness landscape with Gaussian processes. Proc Natl Acad Sci 110(3):E193–E201
61. Zhou P, Tian F, Chen X, Shang Z (2008) Modeling and prediction of binding affinities between the human amphiphysin SH3 domain and its peptide ligands using genetic algorithm-Gaussian processes. Pept Sci 90(6):792–802
62. Cortes-Ciriano I, van Westen GJP, Lenselink EB, Murrell DS, Bender A, Malliavin T (2014) Proteochemometric modeling in a Bayesian framework. J Cheminform 6(1):35
63. Bosc N, Atkinson F, Felix E, Gaulton A, Hersey A, Leach AR (2019) Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. J Cheminform 11(1):4
64. Norinder U, Carlsson L, Boyer S, Eklund M (2014) Introducing conformal prediction in predictive modelling. A transparent and flexible alternative to applicability domain determination. J Chem Inform Model 54(6):1596–1603
65. Sun J, Carlsson L, Ahlberg E, Norinder U, Engkvist O, Chen H (2017) Applying mondrian cross-conformal prediction to estimate prediction confidence on large imbalanced bioactivity data sets. J Chem Inf Model 57(7):1591–1598
66. Svensson F, Afzal AM, Norinder U, Bender A (2018) Maximizing gain in high-throughput screening using conformal prediction. J Cheminform 10(1):7
67. Norinder U, Boyer S (2016) Conformal prediction classification of a large data set of environmental chemicals from ToxCast and Tox21 estrogen receptor assays. Chem Res Toxicol 29(6):1003–1010
68. Norinder U, Boyer S (2017) Binary classification of imbalanced datasets using conformal prediction. J Mol Graph Model 72:256–265
69. Svensson F, Norinder U, Bender A (2017) Modelling compound cytotoxicity using conformal prediction and PubChem HTS data. Toxicol Res 6(1):73–80
70. Forreryd A, Norinder U, Lindberg T, Lindstedt M (2018) Predicting skin sensitizers with confidence—using conformal prediction to determine applicability domain of gard. Toxicol In Vitro 48:179–187
71. Cortés-Ciriano I, Bender A, Malliavin T (2015) Prediction of PARP inhibition with proteochemometric modelling and conformal prediction. Mol Inf 34(6–7):357–366
72. Papadopoulos H, Vovk V, Gammerman A (2011) Regression conformal prediction with nearest neighbours. J Artif Intell Res 40:815–840

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.