JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Cite This: J. Chem. Inf. Model. 2019, 59, 2642−2655

Article

pubs.acs.org/jcim

# Building Quantitative Structure−Activity Relationship Models Using Bayesian Additive Regression Trees

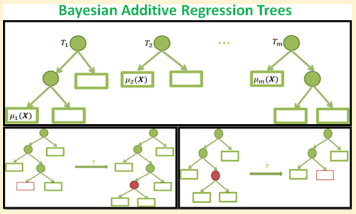
Dai Feng,*[,†,∥] Vladimir Svetnik,*[,†,∥] Andy Liaw,[†] Matthew Pratola,[‡] and Robert P. Sheridan[§]

[†]Biometics Research, Merck & Co., Inc., Kenilworth, New Jersey 07033, United States

[‡]Department of Statistics, The Ohio State University, Cockins Hall, 1958 Neil Avenue, Columbus, Ohio 43210, United States

[§]Modeling and Informatics, Merck & Co., Inc., Kenilworth, New Jersey 07033, United States

**ABSTRACT:** Quantitative structure−activity relationship (QSAR) is a very commonly used technique for predicting the biological activity of a molecule using information contained in the molecular descriptors. The large number of compounds and descriptors and the sparseness of descriptors pose important challenges to traditional statistical methods and machine learning (ML) algorithms (such as random forest (RF)) used in this field. Recently, Bayesian Additive Regression Trees (BART), a flexible Bayesian nonparametric regression approach, has been demonstrated to be competitive with widely used ML approaches. Instead of only focusing on accurate point estimation, BART is formulated entirely in a hierarchical Bayesian modeling framework, allowing one to also quantify uncertainties and hence to provide both point and interval estimation for a variety of quantities of interest. We studied BART as a model builder for QSAR and demonstrated that the approach tends to have predictive performance comparable to RF. More importantly, we investigated BART's natural capability to analyze truncated (or qualified) data, generate interval estimates for molecular activities as well as descriptor importance, and conduct model diagnosis, which could not be easily handled through other approaches.






## INTRODUCTION

In QSAR, a statistical model is generated from a training set of molecules (represented by chemical descriptors) and their biological activities. The model can be used to predict the activities of molecules not in the training set. Such predictions help prioritize the selection of experiments to perform during the drug discovery process. Higher prediction accuracy is always desirable and traditionally pursued by the model developers, but another important aspect of QSAR that is often missed in practice is estimation of the uncertainty in the activity predicted for each molecule. The uncertainty in prediction may have several sources including the following:

1. The original data against which a QSAR model is calibrated (trained) has an error in the activity measurement.
2. A model (equations, algorithms, rules, etc.) and chemical descriptors may be grossly inadequate causing a systematic bias in the prediction.
3. A model is, typically, built on a finite random sample of observations rendering the model parameter estimates to be random variables as well.

The reason why prediction uncertainty is often overlooked is that the development of a statistical methodology for the uncertainty estimation is somewhat lagging behind the QSAR model applications. This is especially true for the highly accurate but quite complex models such as SVM, Random Forest (RF),[1] Boosting,[2,3] and Deep Neural Networks,[4] used by QSAR modelers. Recent research in Machine Learning and Statistics[5−8] resulted in several promising approaches that close the gap between model prediction capability and estimating model uncertainties. For example, conformal regression and classification[9] allow a QSAR modeler to estimate a prediction interval which covers the unknown "true" molecular activity with a selected confidence. This approach is quite general and can be applied to practically any prediction model. One of the limitations of the conformal prediction, however, is the necessity to have separate calibration data in order to estimate the distribution of prediction errors, but in the case of RF[10] this limitation is not essential since so-called out-of-bag samples can be used for calibration. Another approach specific to RF-type models is Quantile Random Forest,[7] where prediction intervals can be constructed directly from the predicted quantiles obtained simultaneously with the prediction of the variable of interest. Of note here is that both conformal (with the modification of nonconformity score[10]) and quantile regression provide prediction intervals conditioned on molecule descriptors, and as such have the interval width reflecting uncertainty in the predicted activity for each molecule. This width, however, reflects only one part of the uncertainty, i.e. uncertainty due to the random error in the measurement of the activity. What is unaccounted for is the uncertainty in the estimated parameters of the model caused by the random selection of the training data. Simply put, if the model were trained on a different training data, the model parameters would be different as would the prediction interval width. Recent work by Wager et al.[8] provides estimates of this uncertainty for the RF models which we will use in this paper.

**Received:** January 29, 2019
**Published:** April 18, 2019




DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

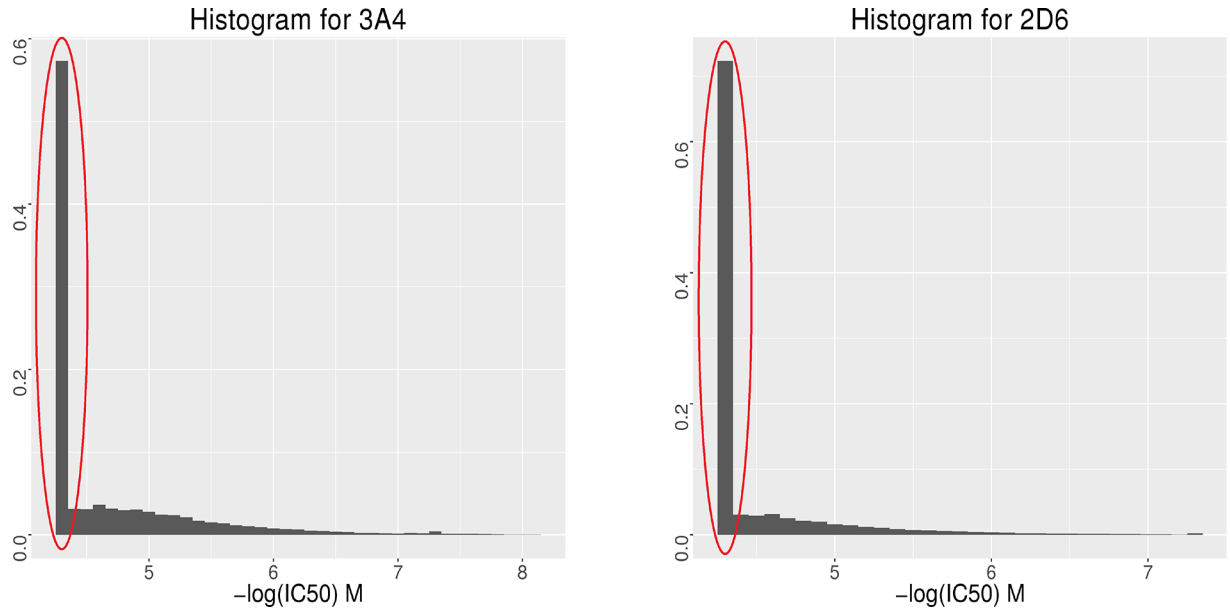



**Figure 1.** Histograms of two truncated data sets.

Methods for estimating uncertainty in the examples above and in general are prediction model specific, adding computational burden to the prediction itself. Bayesian modeling, on the other hand, offers a conceptually simple way for estimating the uncertainty as essentially a byproduct of the modeling process. For example, an unknown function relating chemical descriptors and biological activity can be assumed to be a realization of a Gaussian process (GP).[11−13] Under this assumption, predicted molecular activity has a Gaussian distribution and the point estimate of the activity is equal to the mean of this distribution. According to the GP theory, to calculate the mean, one also needs to calculate the standard deviation of the same distribution, based on which the width of the prediction interval can be obtained. Thus, one essentially obtains both predicted molecular activity and corresponding prediction interval. Unfortunately, the $O(n^3)$ scaling of the computational cost for that method precludes its use in our applications when the number of samples, $n$, is large.

Recently, Bayesian Additive Regression Trees (BART), a flexible Bayesian nonlinear regression approach developed in Chipman et al.,[14] have been demonstrated to be competitive with widely used Machine Learning models. BART is formulated in a Bayesian hierarchical modeling framework and as such provides both predictions and prediction interval estimates for a variety of quantities of interest. We studied BART as a model builder for QSAR and demonstrated that the approach tends to have predictive performance comparable to RF. More importantly, we investigated BART's natural capability to analyze truncated (or qualified) data, generate interval estimates for molecular activities as well as descriptor importance, and conduct model diagnosis, which could not be easily handled through other approaches.

## METHODS

**Data Sets.** The same 30 data sets as used in Ma et al.[4] were used in this study. They are in-house MSD data sets including on-target and ADME (absorption, distribution, metabolism, and excretion) activities. Among the 30 data sets, 15 are the same data as were used for the Kaggle competition;[4] a separate group of 15 different data sets were used to further compare the performance of the different methods. Training and test data for all 30 data sets (with disguised molecule names and descriptors) are publicly available through the Supporting Information of the paper by Sheridan et al.[3]

There are several properties of the data sets that pose important challenges to prediction of molecular activities. First, the size of the data can be very large. For example, in the "HERG (full data set)", there are 318,795 molecules and 12,508 descriptors. In this paper, we used a set of descriptors that is the union of AP, the original "atom pair" descriptor from Carhart et al.,[15] and DP descriptors ("donor−acceptor pair"), called BP in the work of Kearsley et al.[16] Second, the number of molecules can be smaller than the number of descriptors—the so-called small $n$ large $p$ problem. For example, for the "A-II" data, there are 2763 molecules but 5242 descriptors. Third, the descriptors are sparse. Among the 30 data sets, on average only 5.2 ± 1.8% of the data are nonzero entries. Fourth, strong correlations may exist between different descriptors. Fifth, there are several qualified data sets in which data were truncated at some threshold values. For example, we might know only that the measured IC50 (concentration that results in 50% inhibition) is greater than 30 $\mu$M because 30 $\mu$M was the highest concentration in the titration. The histograms of two qualified data sets are shown in Figure 1.

Last but not least, a usual way of splitting the data into the training and test sets is by random selection, i.e. "split at random". However, the separation of training and test sets in this study was obtained through a "time-split". In the actual practice in a pharmaceutical environment, QSAR models are applied prospectively. Predictions are made for compounds not yet tested in the appropriate assay, and these compounds may or may not have analogs in the training set. In this study, for each data set, the first 75% of the molecules assayed for the particular activity formed the training set, while the remaining 25% of the compounds assayed later formed the test set. The key to building a high-performance machine learning model/algorithm is to train the model/algorithm on and test it against data that come from the same target distribution. Using "time-split", however, since training and test sets were not randomly selected from the


DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

same pool of compounds, the data distributions in these two subsets are frequently not the same, or even similar to each other. This violates the underlying assumption of many machine learning methods and poses a great challenge to them.

**Random Forest.** One of the purposes of this study is to compare BART with RF. Both RF and BART are ensemble-of-trees methods. Both can capture nonlinear structures with complex interaction in high-dimensional data. RF is a bagging method[17] that first builds a large collection of decorrelated trees on bootstrap samples and then averages them. BART, using the similar idea as gradient boosting,[18] models the data by a cumulative effort of trees as weak leaners. They can both handle regression and classification problems.

RF is an ensemble of $B$ trees $T_1(\boldsymbol{X}), ..., T_B(\boldsymbol{X})$, where $\boldsymbol{X} = (x_1, ..., x_p)$ is a $p$-dimensional vector of molecular descriptors or properties of a molecule. Let $D = (\boldsymbol{X}_1, Y_1), ..., (\boldsymbol{X}_n, Y_n)$, where $\boldsymbol{X}_i$, $i = 1, ..., n$, is a vector of descriptors and $Y_i$ is either the corresponding class label (e.g., active/inactive) or activity of interest (e.g., −log IC50).

For each tree, define $L_b(\boldsymbol{X})$ as the set of training examples falling in the same "leaf" as $\boldsymbol{X}$. The weights $w_i(\boldsymbol{X})$ then capture the frequency with which the $i$-th training example falls into the same leaf as $\boldsymbol{X}$:

$$w_i(\boldsymbol{X}) = \frac{1}{B}\sum_{b=1}^{B} w_{bi}(\boldsymbol{X})$$

where $w_{bi}(\boldsymbol{X}) = \frac{1(\{\boldsymbol{X}_i \in L_b(\boldsymbol{X})\})}{|L_b(\boldsymbol{X})|}$ and $1(\cdot)$ is an indicator function.

RF estimates the conditional mean activity of a molecule, $E(Y|\boldsymbol{X})$, by the weighted mean over the observations of the response variable $Y$,

$$\hat{\mu}(\boldsymbol{X}) = \sum_{i=1}^{n} w_i(\boldsymbol{X}) Y_i$$

Assume that molecular activity and descriptors are related by the following general model:

$$Y(\boldsymbol{X}) = f(\boldsymbol{X}) + \epsilon(\boldsymbol{X})$$

where $f(\boldsymbol{X})$ is an unknown function which we approximate by RF, and $\epsilon(\boldsymbol{X})$ is a random error. Suppose that $\hat{Y}(\boldsymbol{X}^*)$ is the point estimate, i.e. the RF predicted activity of a new molecule with descriptor vector $\boldsymbol{X}^*$, and the "true" unknown activity is $Y(\boldsymbol{X}^*)$. A prediction interval (PI) is an interval that covers $Y(\boldsymbol{X}^*)$ with a prespecified probability, e.g. equal to 95%, and as such is an interval estimate quantifying uncertainty in the RF prediction.

To estimate PI, we use the well-known bias-variance decomposition for the expected squared error in regression:

$$\begin{aligned} &E_{D,\epsilon}(Y(\boldsymbol{X}^*) - \hat{Y}(\boldsymbol{X}^*))^2 \\ &\quad = (f(\boldsymbol{X}^*) - E_D(\hat{Y}(\boldsymbol{X}^*)))^2 + E_D \\ &\qquad (\hat{Y}(\boldsymbol{X}^*) - E_D(\hat{Y}(\boldsymbol{X}^*)))^2 + E_\epsilon(Y(\boldsymbol{X}^*) - f(\boldsymbol{X}^*))^2 \end{aligned}$$

where $E$ denotes mathematical expectation and subscripts indicate whether expectations are taken with respect to the random training sample $D$ or random error $\epsilon(\boldsymbol{X})$, or both. The three terms on the right-hand side of the above equation represent respectively, squared bias, variance of the estimate, $Var(\hat{Y}(\boldsymbol{X}^*))$, and variance of the random error, $Var(\epsilon(\boldsymbol{X}^*))$. Note that all expectations are conditional with respect to the descriptor vector $\boldsymbol{X}^*$. In the case of RF, bias in prediction error is typically small, provided that the individual trees have sufficient depth, and as such are approximately unbiased. Omitting the bias term, the expected squared error becomes approximately equal to the variance of the predicted error, $Err(\boldsymbol{X}^*) = Y(\boldsymbol{X}^*) - \hat{Y}(\boldsymbol{X}^*)$, where

$$Var(Err(\boldsymbol{X}^*)) = Var(\hat{Y}(\boldsymbol{X}^*)) + Var(\epsilon(\boldsymbol{X}^*))$$

Substituting the two variances on the right-hand side with their estimates, $\widehat{Var}(\hat{Y}(\boldsymbol{X}^*))$ and $\widehat{Var}(\hat{\epsilon}(\boldsymbol{X}^*))$ discussed later, one obtains an estimate of variance of $Err(\boldsymbol{X}^*)$

$$\widehat{Var}(Err(\boldsymbol{X}^*)) = \widehat{Var}(\hat{Y}(\boldsymbol{X}^*)) + \widehat{Var}(\hat{\epsilon}(\boldsymbol{X}^*))$$

Assuming that $Err(\boldsymbol{X}^*)$ has Gaussian distribution with mean zero (due to the unbiasedness of $\hat{Y}(\boldsymbol{X}^*)$) and variance $Var(Err(\boldsymbol{X}^*))$, one can derive an estimate of the $100(1-\alpha)\%$ prediction interval for $Y(\boldsymbol{X}^*)$ as follows:

$$\begin{aligned} Y(\boldsymbol{X}^*) \in (&\hat{Y}(\boldsymbol{X}^*) - z_{1-\alpha/2}\sqrt{\widehat{Var}(Err(\boldsymbol{X}^*))}\,,\ \hat{Y}(\boldsymbol{X}^*) \\ &+ z_{1-\alpha/2}\sqrt{\widehat{Var}(Err(\boldsymbol{X}^*))}\,) \end{aligned}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-th percentile of the standard normal distribution (e.g., for the 95% PI, $\alpha/2 = 0.025$).

Obtaining the variance of RF prediction, $Var(\hat{Y}(\boldsymbol{X}^*))$, is straightforward if one could use a bootstrap approach whereby the training data $D$ is repeatedly sampled with replacement; for each of these samples a RF estimate is obtained, and the sample variance of these estimates is calculated giving rise to the estimate $\widehat{Var}(\hat{Y}(\boldsymbol{X}^*))$. Note that there are two bootstrap processes: internal and external. The external process provides, say $B_E$, bootstrap samples $D_j$, $j = 1, ..., B_E$, each of which consists of $n$ samples (pairs of molecular activities and corresponding descriptors). The internal process on the other hand, bootstraps $B$ times each of $D_j$ samples to build $B$ RF trees. Thus, estimation of $Var(\hat{Y}(\boldsymbol{X}^*))$ would require $B_E \times B$ bootstrap samples of the original training data $D$ which could be computationally prohibitive.

Recently, Wager et al.[8] proposed approaches where estimation of $Var(\hat{Y}(\boldsymbol{X}^*))$ requires only $B$ bootstrap samples that were used to calculate the RF estimate itself, thus significantly reducing the computational burden of the external bootstrap. The methods proposed in Wager et al. are based on the jackknife and infinitesimal jackknife (IJ). We used the approach based on the IJ, since it was shown that the IJ estimator is more efficient than the jackknife estimator by having a lower Monte Carlo variance. We refer the reader to their paper[8] for the details.

To obtain an estimate of $\mathrm{Var}(\hat{\epsilon}(\mathrm{X}^*))$, we used the following formula:

$$\frac{1}{\sum_{b=1}^{B}|L_b(\boldsymbol{X})| - B} \sum_{b=1}^{B}\sum_{i=1}^{n}(1(\{\boldsymbol{X}_i \in L_b(\boldsymbol{X})\})(y_i) - w_{bi}(\boldsymbol{X})(y_i))^2$$

We refer to this method as IJRF hereafter.

To obtain PIs, another method proposed is the quantile regression forests (QRF) method of Meinshausen (2006),[7] which is a generalization of random forests (RF).[17] The QRF provides estimates of conditional quantiles for high-dimensional predictor variables. Similarly to the conditional mean $E(Y|\boldsymbol{X})$, we can approximate


DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

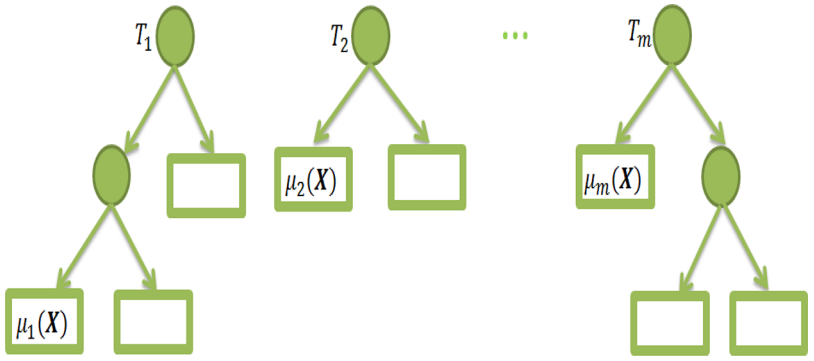



**Figure 2.** Schematic illustration of the BART model.

$$F(y|\boldsymbol{X}) = P(Y \le y|\boldsymbol{X}) = E(I_{\{Y \le y\}}|\boldsymbol{X})$$

by

$$\hat{F}(y|\boldsymbol{X}) = \sum_{i=1}^{n} w_i(\boldsymbol{X}) I_{\{Y_i \le y\}}$$

Note that, in general, the $\alpha$-th quantile, $Q_\alpha(x)$, is defined as

$$Q_\alpha(\boldsymbol{X}) = \inf\{y: F(y|\boldsymbol{X}) \le \alpha\}$$

The quantiles provide more comprehensive information about the distribution of $Y$ as a function of descriptors/features $\boldsymbol{X}$ than the conditional mean alone.

Quantile regression can be used to build prediction intervals. The $\alpha/2$ and $1-\alpha/2$-th quantiles give the lower and upper bounds of a corresponding $100(1-\alpha)\%$ interval, respectively. For example, a 95% prediction interval for the value of $Y$ is given by $(Q_{025}(x), Q_{975}(x))$. We refer to this method as QRF hereafter.

Furthermore, for a normal distribution,

$$Q_{0.75} - Q_{0.25} \approx 1.35\sigma \tag{1}$$

where $\sigma$ is the standard deviation of the normal distribution. Assuming data are normally distributed, we can use the quantile regression to estimate $Q_{0.75}$ and $Q_{0.25}$ and then obtain the estimate of $Var(\epsilon(\boldsymbol{X})) = \sigma^2$ based on eq 1. We refer to this method as IJQRF hereafter.

**Bayesian Additive Regression Trees.** The Bayesian additive regression tree (BART)[14] uses a sum of trees to approximate $E(Y|\boldsymbol{X})$. By weakening the individual tree effects, BART ends up with a sum of trees, each of which explains a small and different portion of the underlying function. In other words, BART, using a similar idea as gradient boosting,[18] models the data by a cumulative effort with each stage introducing a weak learner (tree $T_i(\boldsymbol{X})$ at stage $i$) to compensate the shortcomings of existing weak learners (trees $T_1(\boldsymbol{X})$, ..., $T_{i-1}(\boldsymbol{X})$). The model combines additive and interaction effects. It also regularizes the fit by keeping the individual tree effects small (via a penalty prior as explained in detail below).

The BART model is as follows:

$$Y(\boldsymbol{X}) = \mu_1(\boldsymbol{X}) + \mu_2(\boldsymbol{X}) + \ldots + \mu_m(\boldsymbol{X}) + \sigma z,$$
$$z \sim N(0,1)$$

where $\mu_i(\boldsymbol{X})$ is the mean in the bottom node where $\boldsymbol{X}$ falls to in the $i$-th tree. A schematic illustration of the BART model is shown in Figure 2.

In contrast to other tree-based methods which are usually algorithm-based, BART is formulated entirely as a Bayesian hierarchical model, which provides great flexibility and power for data analysis.[19] It is fully characterized by the following three components: a likelihood function, a collection of unknown parameters, and a prior distribution over these parameters. The Bayesian framework allows quantifying uncertainties and hence provides both point and interval estimation for a variety of quantities of interest, for example the credible interval of variable importance, a goal not easily achieved by other tree-based methods. Furthermore, within this framework, practical issues such as the handling of truncation of qualified data and model diagnosis can all be accommodated.

The likelihood function is as follows:

$$Y_j \Big| \boldsymbol{X}_j \overset{\text{iid}}{\sim} N\left(\sum_{i=1}^{m} \mu_i(\boldsymbol{X}_j), \sigma^2\right)$$

The unknown parameters are

- The trees $T_1(\boldsymbol{X})$, ..., $T_m(\boldsymbol{X})$.
- The node parameters $M_i = (\mu_{i1}, ..., \mu_{i\delta i})$ for each tree $i = 1$, ..., $m$.
- The residual standard deviation $\sigma$.

The prior on tree $T_i(\boldsymbol{X})$ is specified through a branching process.[20] The probability a current bottom node, at depth $d$, gives birth to a left and right child is

$$\frac{\alpha}{(1+d)^\beta}$$

The usual BART defaults are $\alpha = 0.95$ and $\beta = 2$. This specification makes non-null but small trees likely. To construct a tree, at each branching, first a descriptor is drawn uniformly from all descriptors and then a cut-point is drawn uniformly from the discretized range of the drawn descriptor.

For the prior on means $\mu_i = \mu_i(\boldsymbol{X})$, let

$$\mu_i \sim N(0, \tau^2), \text{ i. i. d.}$$

Note that a priori,

$$E(Y) = \sum_{i=1}^{m} \mu_i \sim N(0, m\tau^2)$$

Then $\tau$ is chosen by centering the data and assuming $E(Y) \in (y_{min}, y_{max})$ with high probability. For instance, setting

$$\tau = \frac{y_{\max} - y_{\min}}{2k\sqrt{m}}$$



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

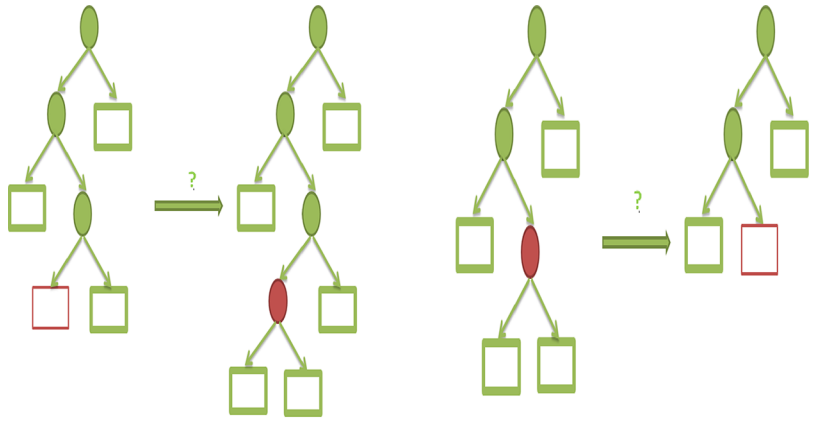



**Figure 3.** Illustration of birth and death step in updating trees.

and taking $k = 2$ implies a 95% prior probability that $E(Y)$ lies in the observed range of the data, while taking $k = 3$ implies that a 99.7% prior probability. The recommended default value of $k$ is 2.

For the prior on $\sigma$, let

$$\sigma^2 \sim \frac{\nu\lambda}{\chi_\nu^2}$$

where the default value of $\nu$ is 3. As to $\lambda$, first get a reasonable estimate of $\hat{\sigma}$ of $\sigma$ and then choose $\lambda$ to put $\hat{\sigma}$ at a specified quantile of the $\sigma$ prior. We take $\hat{\sigma}$ equal to the least-squares estimate if $p < n$; otherwise, it equals to $sd(Y)$. The default quantile is 0.9.

To obtain the posterior distribution on all unknown parameters that determine the sum-of-trees model,

$$p(T_1, M_1, \ldots, T_m, M_m, \sigma|Y)$$

a Metropolis within Gibbs Markov Chain Monte Carlo (MCMC) sampler[21] is used in BART.[14] At each iteration, a tree $T_i$ and its corresponding means $M_i$ are updated given the other parameters based on the following conditional distribution.

$$p(T_i, M_i|T_1, M_1, \ldots, T_{i-1}, M_{i-1}, T_{i+1}, M_{i+1}, \ldots, T_m, M_m, \sigma)$$

The residual standard deviation $\sigma$ is updated based on the conditional distribution

$$p(\sigma|T_1, M_1, \ldots, T_m, M_m)$$

Note that to update $(T_i, M_i)$ for one single tree each time, first $M_i$ is integrated out from the joint conditional distribution to draw $T_i$ and then $M_i$ is drawn given $T_i$. This simplifies computation by avoiding the reversible jumps between continuous spaces of varying dimensions.[20]

BART updates a tree by a Metropolis–Hastings (MH) sampler through various moves, among which is the key birth or death proposal.[20] In a birth step, a more complex tree is proposed. In a death step, a simpler tree is proposed. See Figure 3 for an illustration. Each proposal is either accepted or rejected by a certain probability.

BART obtains the prediction and corresponding PI of a molecular activity through the posterior distribution of predictive activity. The posterior distribution is a summary of results from all MCMC iterations. In each iteration, predictions arise from a normal distribution with mean $\sum_{i=1}^{m} \mu_i(\boldsymbol{X})$ and standard deviation $\sigma^2$. The mean from the posterior distribution is used as the prediction, and the corresponding quantiles from the posterior are used to construct the PI.

The Metropolis–Hastings proposals used in the original BART algorithm could lead to poor mixing of the MCMC sample since they do not facilitate efficient traversal of the model space (suffering from local mode stickiness). The consequence of poor mixing is overfitting the data and under-representing model uncertainty. To improve mixing, a "radical restructure" move, which proposes a large change in tree structure without changing the number of leaves nor the partition of observations into leaves, was proposed.[22] However, this approach could not scale up well to handle high-dimensional data. Recently, Pratola[23] proposed more efficient MH sampling algorithms (OpenBT). The first is a rotation proposal that allows for efficient traveling to disparate regions of high likelihood in tree space through local moves. The second is a perturbation proposal that uses more efficient rules to choose splitting variables and cut-points. OpenBT-bart implements BART with these more efficient sampling algorithms.

In the original BART model, all observations follow a normal distribution with the same standard error. A novel heteroscedastic BART model (Pratola et al.)[24] was developed to alleviate this constraint. In OpenBT-hbart, the conditional mean is still modeled as a sum of trees, each of which determines a contribution to the overall mean; the conditional variance is modeled with a product of trees, each of which determines a contribution to the overall variance.

**How To Handle Truncated Data.** The "true" values of data truncated at $Y_t$ are all set at $Y_t$. Since the smallest value of observed activities in the training set is $Y_t$, the lower bound of a confidence interval from QRF has to be larger than or equal to $Y_t$. Therefore, the PI could not cover the "true" value $Y_t$ for those observations that are truncated.

Using BART, the predictive value could be less than the minimum observed value, $Y_t$. Furthermore, we can recover the underlying distribution with no truncation through imputation of missing true activities. In each MCMC iteration, for a truncated activity, we impute the missing truth, assuming the truth follows a truncated normal distribution still with mean $\sum_{i=1}^{m} \mu_i(\boldsymbol{X})$ and standard deviation $\sigma^2$ but bounded above at $Y_t$. The illustration of the issue of truncated data and recovery of underlying distribution by imputation is shown in Figure 4.



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

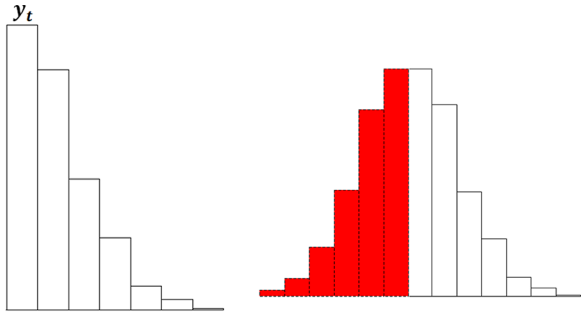



**Figure 4.** Illustration of the issue of truncated data and recovery of underlying distribution by imputation.

For example, in the 3A4 data set (see Figure 1), 55.85% of −log(IC50) values from the training data were truncated at 4.3003. When training the model, instead of using the original observed activities (−log(IC50)), if the observed value equaled the minimum value 4.3003, then we imputed the underlying untruncated value by simulating a value from a normal distribution with corresponding mean of a molecule and standard deviation of the whole training set. Note that different molecules could have different means, although they were all truncated at 4.3003. Due to truncation, the simulated value needed to be less than 4.3003. Furthermore, we imputed the truncated data repeatedly in each MCMC iteration based on the updated mean and standard deviation in each iteration.

## RESULTS

**Comparing RF and BART for Point and PI Estimation.** The metrics to evaluate prediction performance include $R^2$, coverage probability of 95% PIs, and median width of 95% PIs. The $R^2$ is the squared Pearson correlation coefficient between predicted and observed activities in the test set. The same metric was used in the Kaggle competition. For each test set, we calculated the coverage probabilities of nominal 95% PIs of all predicted activities. To obtain coverage probability for each test data set, we computed first the 95% PI for each molecule and then the percentage of how many intervals covered the true activities. The closer the value is to 0.95, the better the result. Besides the coverage probabilities, we compared also the median width of PIs. An ideal case is that an accurate coverage is not offset by a much wider interval.

To run BART, the tuning parameters can be divided into three categories: (1) the hyperparameters $k$ and $\tau$ for the prior on means $\mu_i = \mu_i(\boldsymbol{X})$, the hyperparameters $\nu$ and $\lambda$ for the prior on standard deviation $\sigma$; (2) the number of trees $m$ and $\alpha$ and $\beta$ for the probability to further split a bottom node; (3) the MCMC related parameters including the number of burn-in and number of posterior draws after burn-in.

For the QSAR data, the results were robust to the choice of hyperparameters for means and standard deviation. We set these at default values. Furthermore, the parameter $\alpha$ which decides the probability of splitting a node at root (depth zero) was set at the default.

There could be a relatively large difference in prediction accuracy as the number of trees and depth of each tree vary. To search the best set of two tree related parameters $m$ and $\beta$, we conducted a grid search, using 15 Kaggle data sets, over the values $m$ = (200, 500, 1000, 1500, 2000, 2500) and $\beta$ = (0.5, 1, 1.5, 2). In general, the more complex the model with larger number of deeper trees (larger $m$ and smaller $\beta$), the larger the average value of $R^2$. In addition, the gains are marginal when using too complex models. Zoomed in on each data set, building

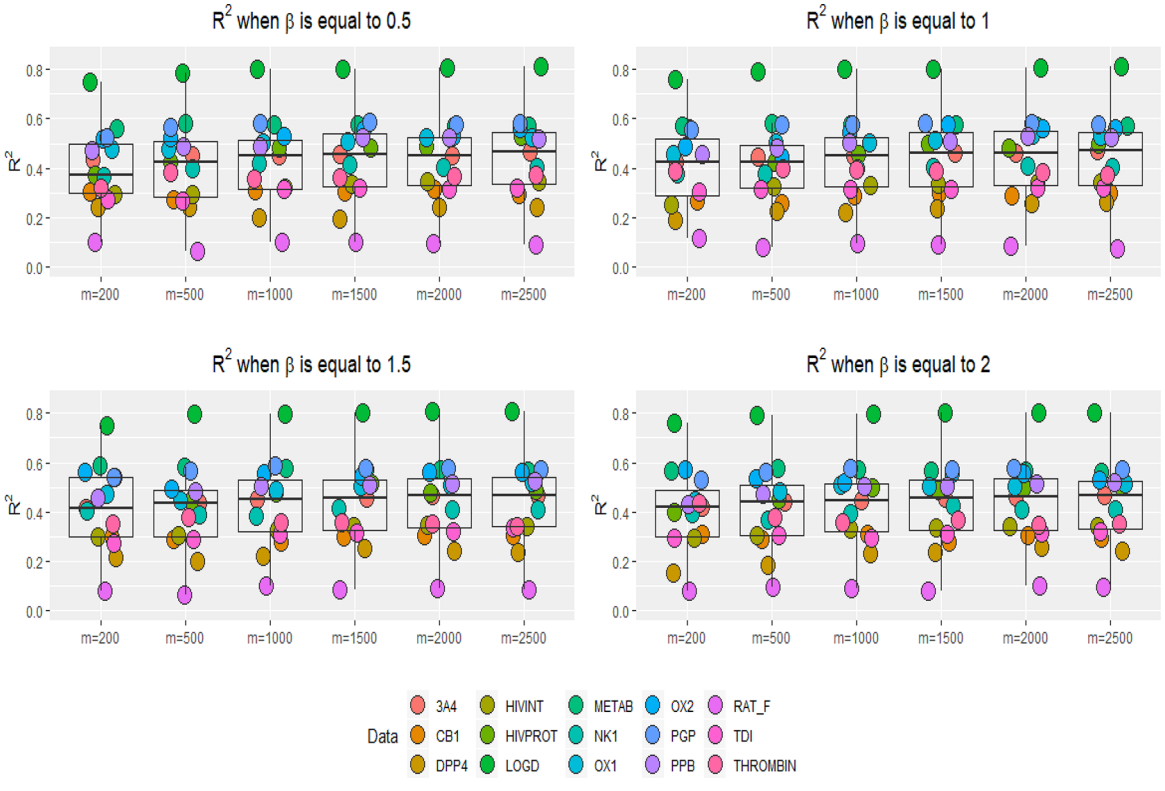



**Figure 5.** $R^2$'s for different combinations of parameters $m$ and $\beta$.



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

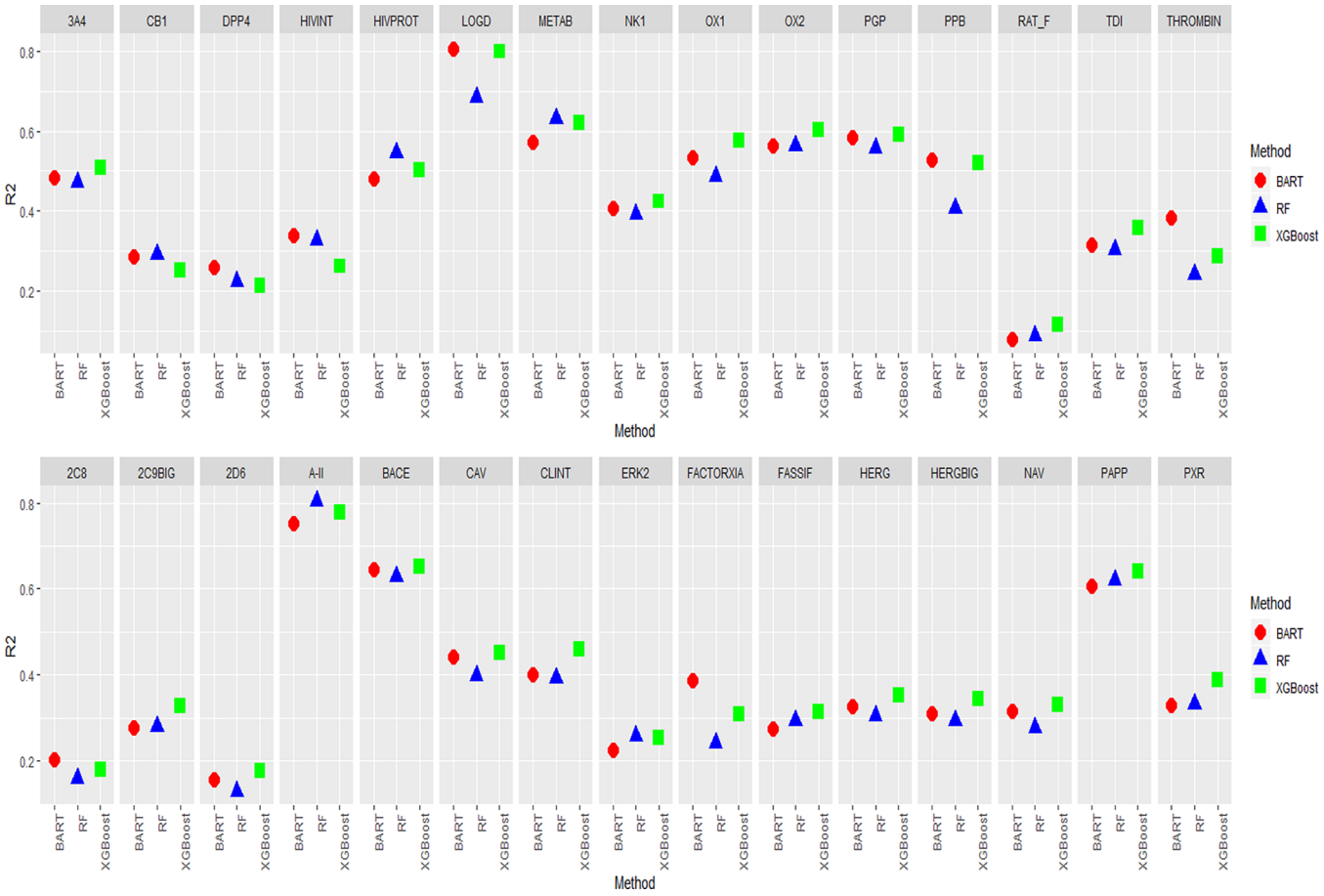



**Figure 6.** $R^2$ for different data sets using different methods.

a more complex model tended to provide larger $R^2$ values and especially so when increasing the number of trees. The more complex the model, the larger the computational cost. To balance the cost and gain of accuracy, we first set $m = 2000$ and then chose $\beta = 1$; this setup generally performed better than others. See Figure 5 for detailed results.

We set the number of burn-in as 1000 and the number of posterior draws after burn-in at 2000. No convergence failure was detected.

To obtain more accurate estimates of standard deviations and hence interval estimations, we also investigated the OpenBT-bart and OpenBT-hbart approaches. The computational cost of OpenBT-bart and OpenBT-hbart is much higher compared with that of BART due to the more complicated MH proposals and model on standard deviation. For the OpenBT-bart, we set the number of trees for mean model at 200, the number of burn-in at 100, the number of posterior draws after burn-in at 1000, and the parameter $\beta$ at 2. The other parameters were set at their defaults. For OpenBT-hbart, the number of trees for the variance model was 40, and the other parameters were the same as that in OpenBT-bart. Note that these setups achieved significant improvement in coverage of PIs as shown below.

For all RF models, following the setup in Ma et al. (2015),[4] we generated 100 trees with $p/3$ descriptors used at each branch point, where $p$ is the number of unique descriptors in the training set. Tree nodes with 5 or fewer molecules were not split further. We applied these parameters to every data set.

Given the underlying similarities between BART and XGBoost, we also considered the point estimate results from XGBoost following the setup in Sheridan et al.[3] The parameters used are as follows. The maximum depth of a tree is 7; the subsample ratio of columns when constructing each tree is 0.56; the "eta" parameter controlling the learning rate is 0.05; the maximum number of iterations is 745; the other parameters are as default.

The different $R^2$'s obtained by RF, BART, and XGBoost for different data sets are shown in Figure 6. The mean $R^2$ for RF, BART, and XGBoost is 0.39, 0.41, and 0.42, respectively.

It is important to assess the uncertainty of not only a prediction but also a performance metric.[25,26] To evaluate the uncertainty of $R^2$, we bootstrapped the data (both training and test) 100 times to calculate the standard deviation of $R^2$. The results are shown in Figure 7. The mean standard deviation for RF, BART, and XGBoost is 0.020, 0.024, and 0.023, respectively.

The different coverage probabilities for different data sets are shown in Figure 8. The mean coverage for QRF, IJRF, IJQRF, BART, OpenBT-bart, and OpenBT-hbart is 0.77, 0.97, 0.98, 0.87, 0.92, and 0.93, respectively.

The different median widths for different data sets are shown in Figure 9. To put median widths on the same scale and compare them to each other, within each data set, we divided each median width by the minimum width.

Comparing different methods, the QRF and BART provided average coverage quite smaller than the nominal (especially so for the QRF), and the coverage probabilities from IJRF, IJQRF, OpenBT-bart, and OpenBT-hbart are much closer to the truth. Compared to OpenBT-bart and OpenBT-hbart, the higher


DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

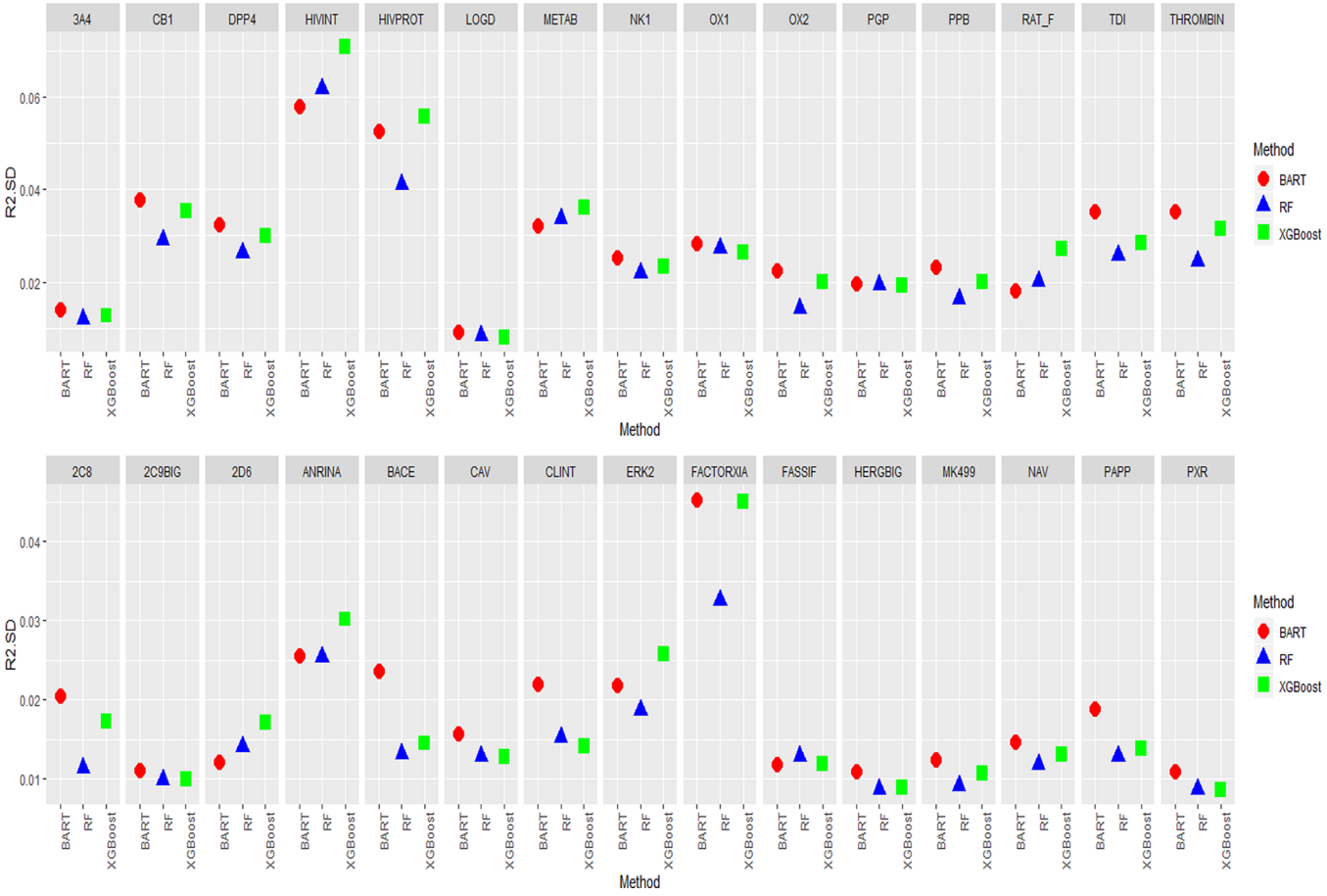



**Figure 7.** Standard deviation of $R^2$ for different data sets using different methods.

coverage of the IJRF and IJQRF generally was offset by a wider and even much wider PI.

Note that for the truncated data, for example, "3A4" and "2D6", the imputation of missing values was implemented in BART. With the attempt of recovering the missing true activities, the standard deviation increased compared to without imputation and hence the width of PI became obviously larger than that from OpenBT-bart and OpenBT-hbart.

**Descriptor Importance.** Both RF and BART can be used to select "important" descriptors. For RF, every descriptor in the out-of-bag (OOB) data is randomly permuted, one at a time, and each modified data set is also predicted by the tree. The difference of squared prediction errors between with and without permutation in the OOB data is calculated for each descriptor and used to evaluate descriptor importance. The larger the difference, the more important the descriptor. For BART, the percentage of a descriptor used in a tree decision rule over all trees (frequency of a descriptor appeared at branch points) is used to evaluate descriptor importance. The larger the percentage, the more important the descriptor.

To investigate the capability of choosing important descriptors by BART and RF, we first conducted a simulation study. To simulate the truth, we use the "LOGD" training data. First, we ran BART using all descriptors to find the first 50 most important ones. Second, we reran the BART using just the 50 descriptors picked. Third, we used the prediction in the rerun as true activities; the 50 descriptors as true important ones; and the reordered importance as the true order of importance for the 50 descriptors. Finally, we simulated 100 data sets each having additional 500 noisy descriptors, which were randomly chosen from other descriptors and permuted. We obtained the truth using RF in a similar way as well.

For simulated data, we ran both BART and RF to select important descriptors and then calculated the percentage of times, among all simulated data sets, that the first 10, 20, 30, 40, and 50 important descriptors chosen were from the corresponding truth. The results are shown in Figure 10.

From the results, first, both BART and RF can screen out the noise—the first 50 important descriptors picked almost never included noise no matter if the truth was from BART or RF. Second, the true order of important descriptors was generally preserved especially when the method used was the same as in the generation of truth.

An additional piece of information on descriptor importance BART can provide, besides the point estimate, is the interval estimation. Figure 11 displays the point estimate and corresponding 90% interval of the first 50 important descriptors in order chosen for LOGD data. The first important descriptor is obviously more important than the others given that there is no overlap between its interval and the intervals of other descriptors (the lower bound of its interval is even larger than the upper bounds of other intervals). The next three important descriptors picked seemed to be relatively important as well among the first 50, especially compared with the last 15 descriptors.

Another observation we had was that the descriptor importance could be misleading when the prediction error was high. In other words, if the model cannot predict accurately, descriptor importance may not be useful. For example, for two


DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

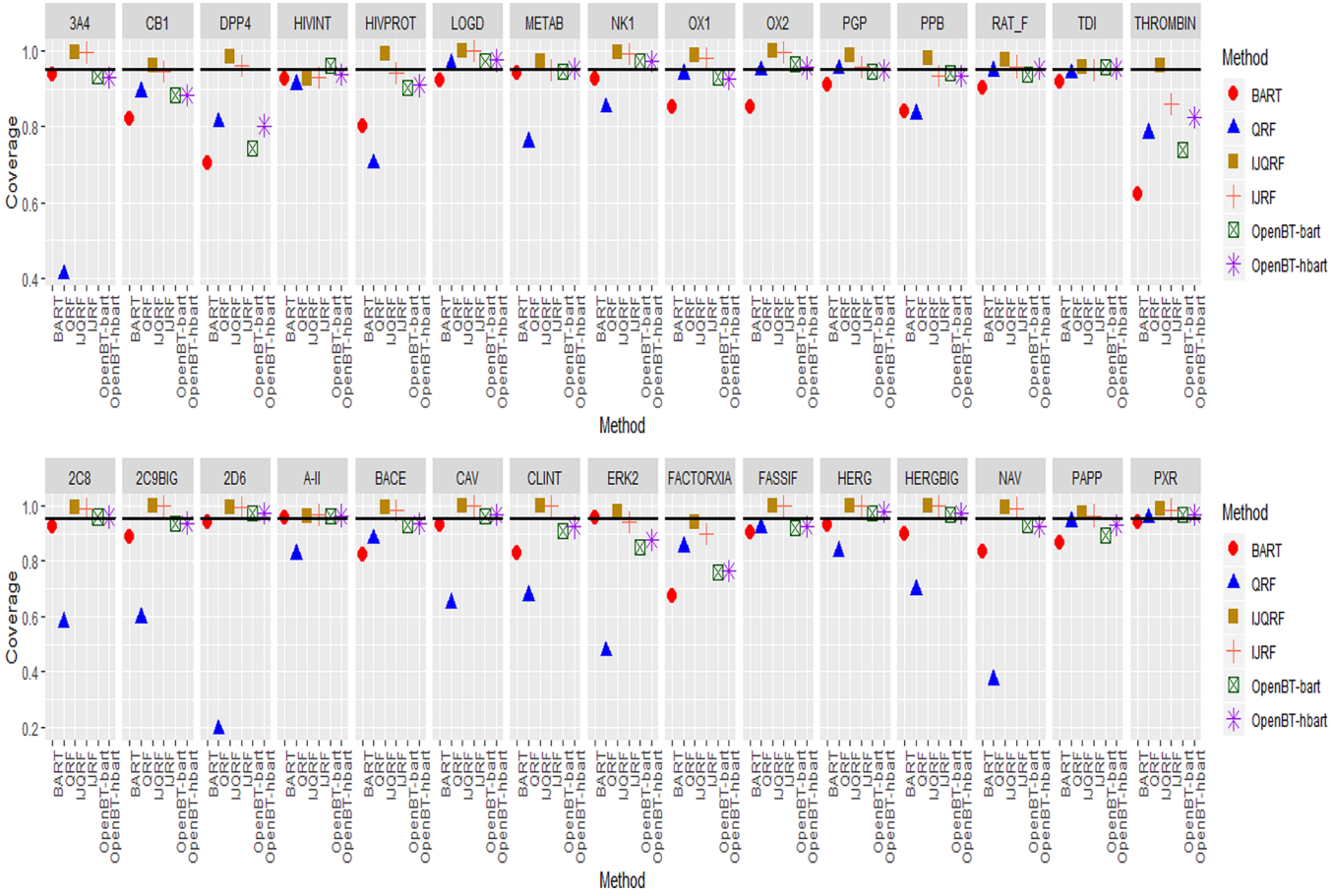



**Figure 8.** Coverage probabilities for different data sets using different methods. The black vertical line in each panel marks the nominal coverage 95%.

data sets "LOGD" and "RAT_F" with high and low $R^2$, we removed the important descriptors chosen (first 10, 20, 30 important descriptors and so on) by either RF and BART and retrained the data, made the prediction, and obtained corresponding $R^2$ again. The results are shown in Figure 12. For the "LOGD" data, removing the important descriptors did reduce the $R^2$. For the "RAT_F", however, the $R^2$ had no obvious trend as we increased the number of important descriptors removed. We suggest to remove important descriptors and then retrain and repredict to confirm the results of descriptor importance.

Interestingly we found that for the "RAT_F" data, the model had a hard time picking important variables or it maybe was the case that every descriptor is nonimportant under the specified BART model. See Figure 13 for details.

**BART Model Diagnostics.** The specified BART model cannot tell which variables are more important for the RAT_F data. A natural question is then: What could go wrong or did the model fit the data well? To study model adequacy, in the Bayesian framework, we can use the posterior predictive checking (PPC).[19] We simulated data repeatedly under the fitted BART model and then compared them to the observed data to check whether there are systematic discrepancies between real and simulated data.

A metric used in PPC is posterior predictive $p$-values: the probability that the replicated data (simulated from posterior distribution) could be more extreme than the observed data

$$p_B = P(T(y^{rep}, \theta) \geq T(y, \theta)|y)$$
$$= \int\int I_{T(y^{rep},\theta)\geq T(y,\theta)} p(y^{rep}|\theta)p(\theta|y)dy^{rep}d\theta$$

Figure 14 exhibits the posterior predictive distribution of the maximum activity, $y_{max}$, obtained from simulated data from the posterior distribution, and the corresponding observed value, $y_{max}observed$, in the training set for "LOGD" and "RAT_F" data, respectively. For the former, there was no potential failing of the model found from the perspective of $y_{max}$—the observed value is about the same as the median of the posterior distribution. For the latter, however, the model did not provide an adequate fit in the tail—the minimum value of $y_{max}$ from the posterior distribution based on the model is much larger than the observed value. The posterior predictive $p$-value was 0.51 and 1, respectively, for "LOGD" and "RAT_F" data. The diagnostic indicated the lack of fit of the BART model for "RAT_F" data and the model needs amendment.

**Software.** The R package *randomForestCI*[27] implemented the proposed method to estimate $Var(\hat{Y}(X^*))$ in Wager et al.[8] The R package *quantregForest*[28] implemented the QRF method proposed in Meinshausen (2006).[7] The BART algorithm based on Chipman et al.[14] was implemented in R packages *BayesTree*, *dbarts*, and *pgbart*. The three packages provided very similar results. One difference among them is that the construction of the model (training) and prediction for a new data set (testing) cannot be separated in *BayesTree*. The *dbarts* package provided the function to separate training from testing by saving all trees starting from version 0.9-0. However, it may need very large memory and therefore cannot accommodate large size QSAR data. The *pgbart* package can also save all the trees and worked


DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

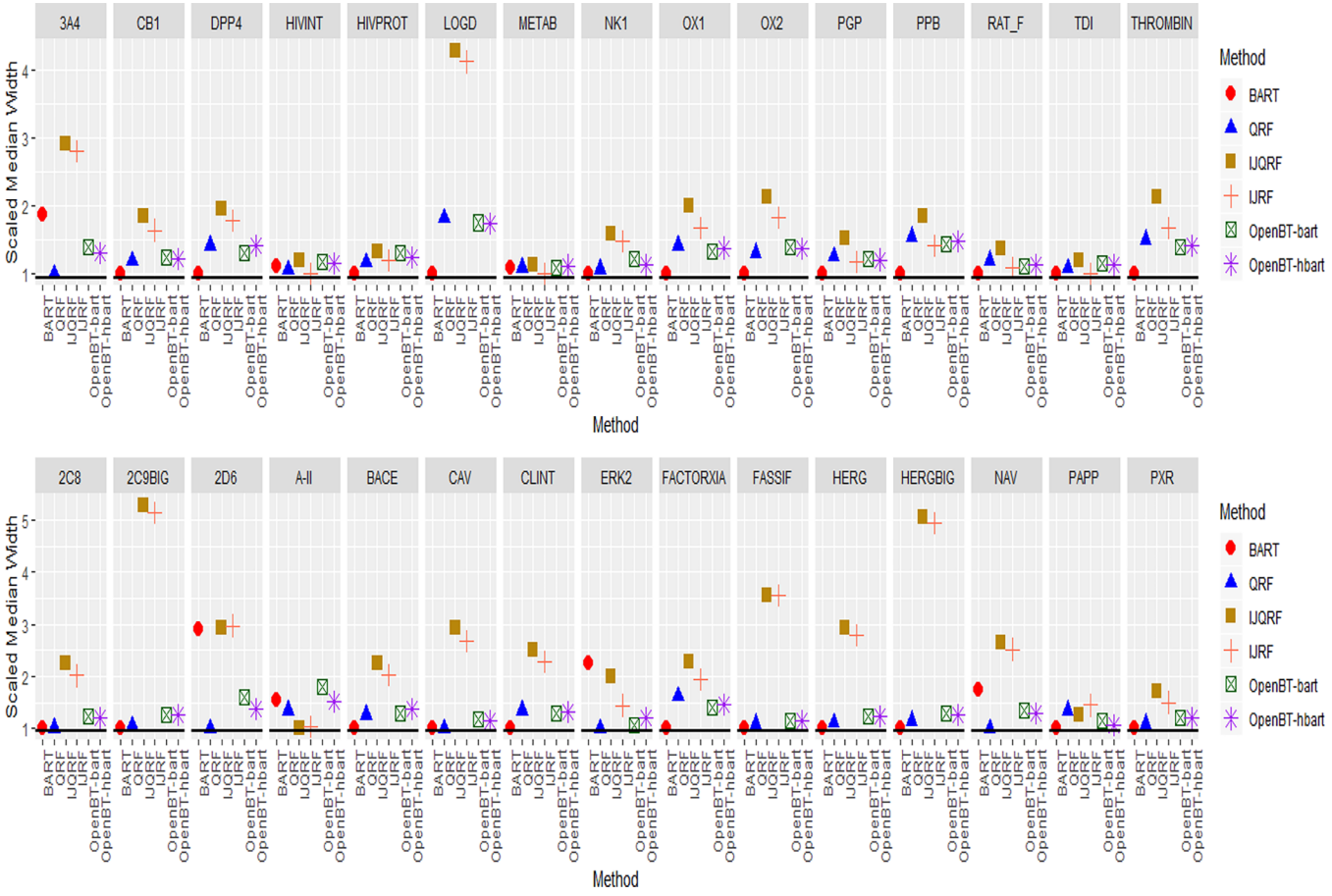



**Figure 9.** Median widths for different data sets using different methods.

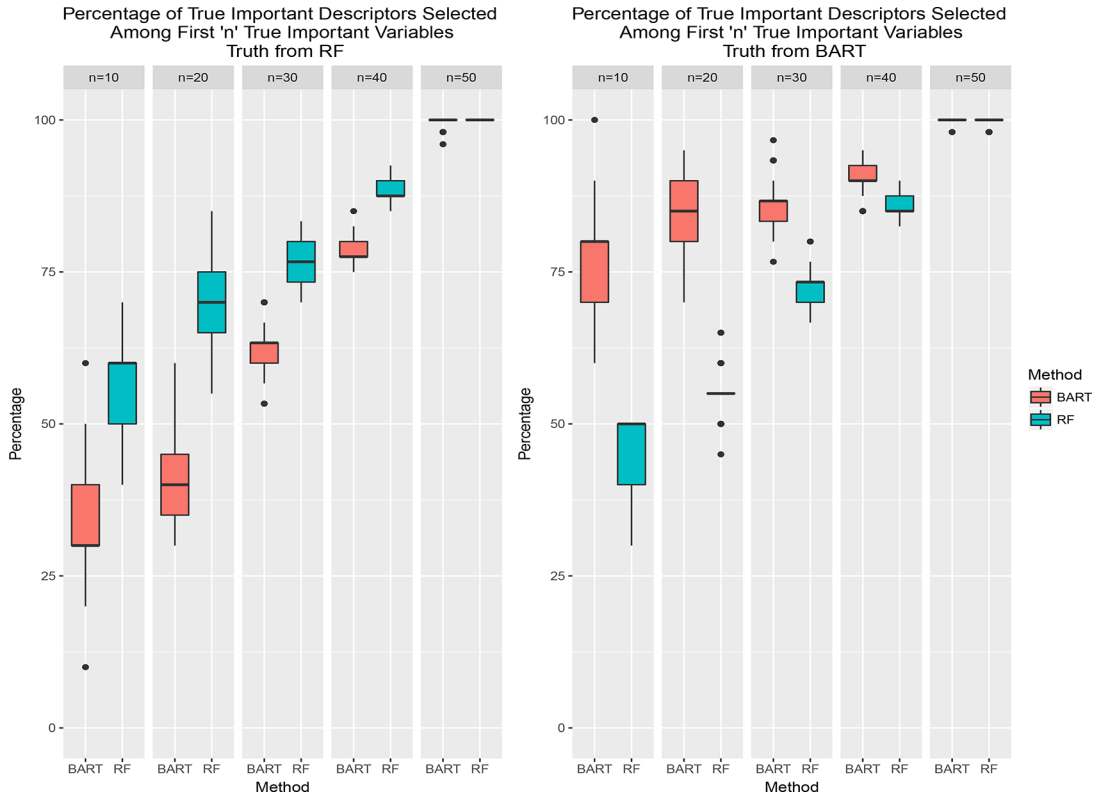



**Figure 10.** Important descriptors picked for simulated data.



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

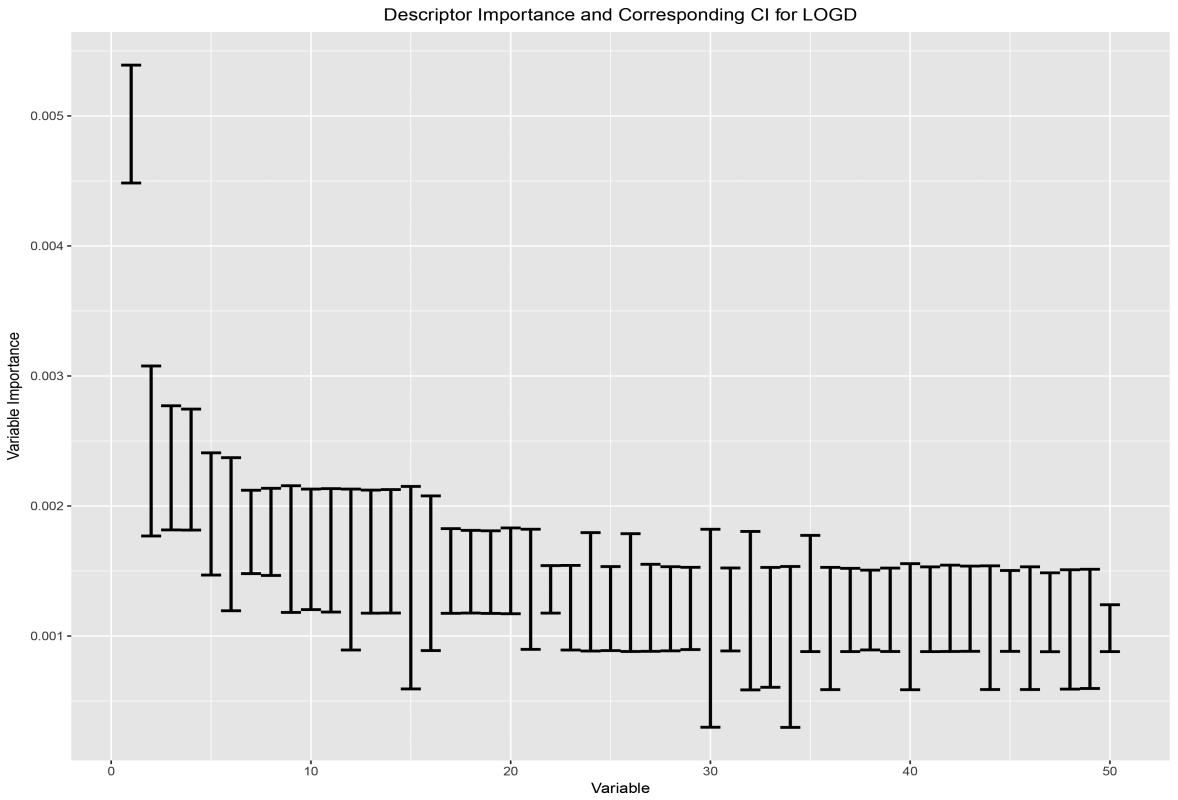



**Figure 11.** Point estimate and corresponding 90% interval of descriptor importance for LOGD data.

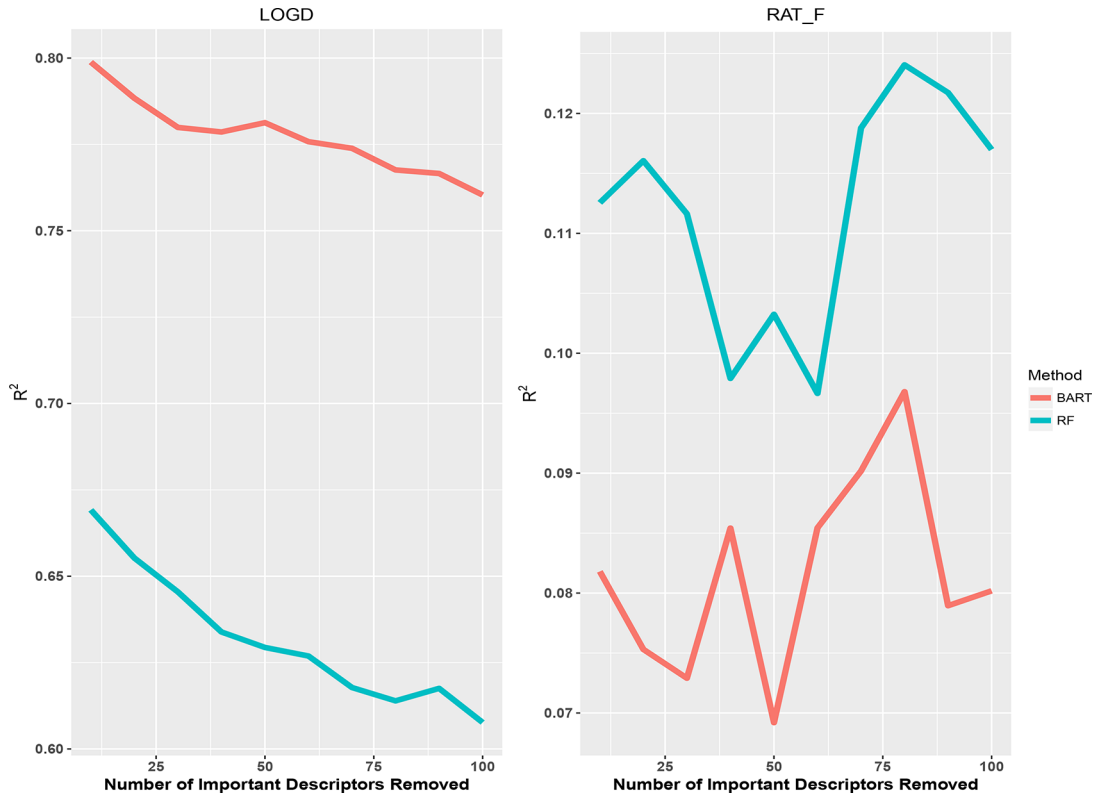



**Figure 12.** $R^2$ after removing "important" descriptors.

fine even for large size QSAR data. The OpenBT algorithms are implemented in an R package available at https://bitbucket.org/mpratola/openbt. The code handling truncation using BART is available from https://bitbucket.org/mpratola/openbt. The code for computing prediction intervals using Random Forest and BART based methods and examples of how to use various approaches are available from https://github.com/Merck/BART-QSAR.



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

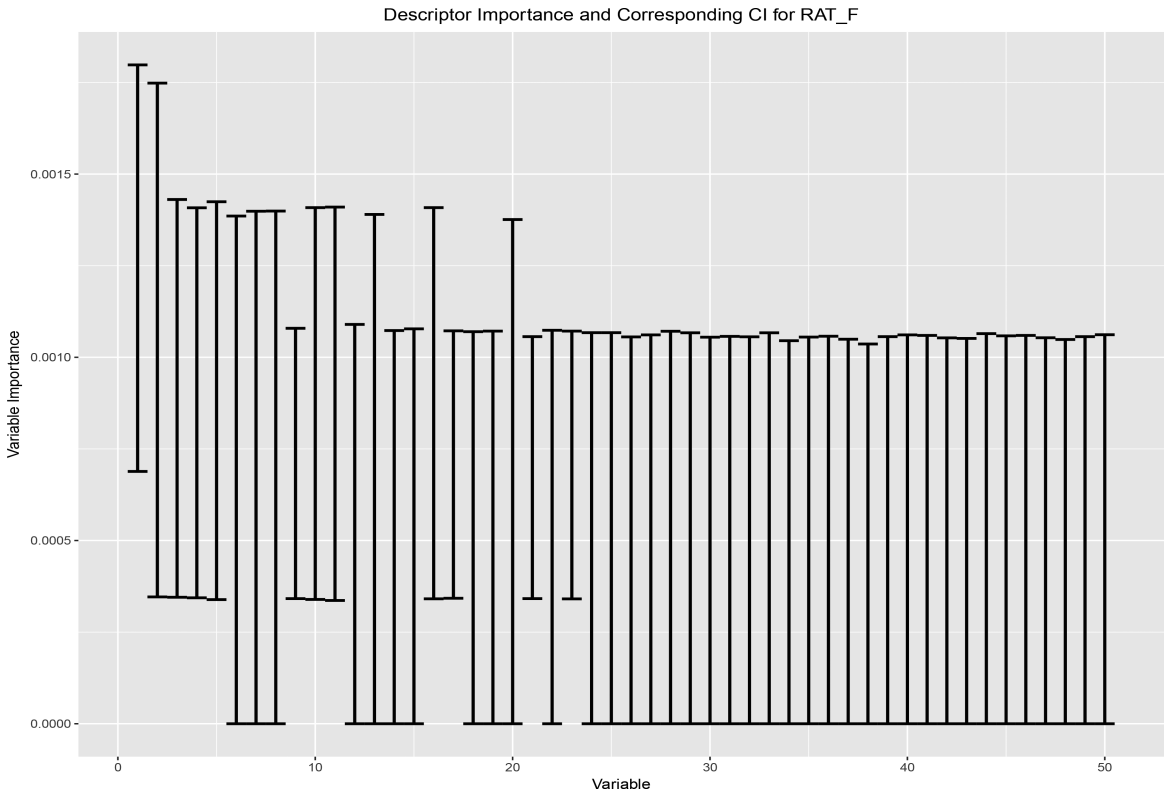



**Figure 13.** Point estimate and corresponding 90% interval of descriptor importance for RAT_F data.

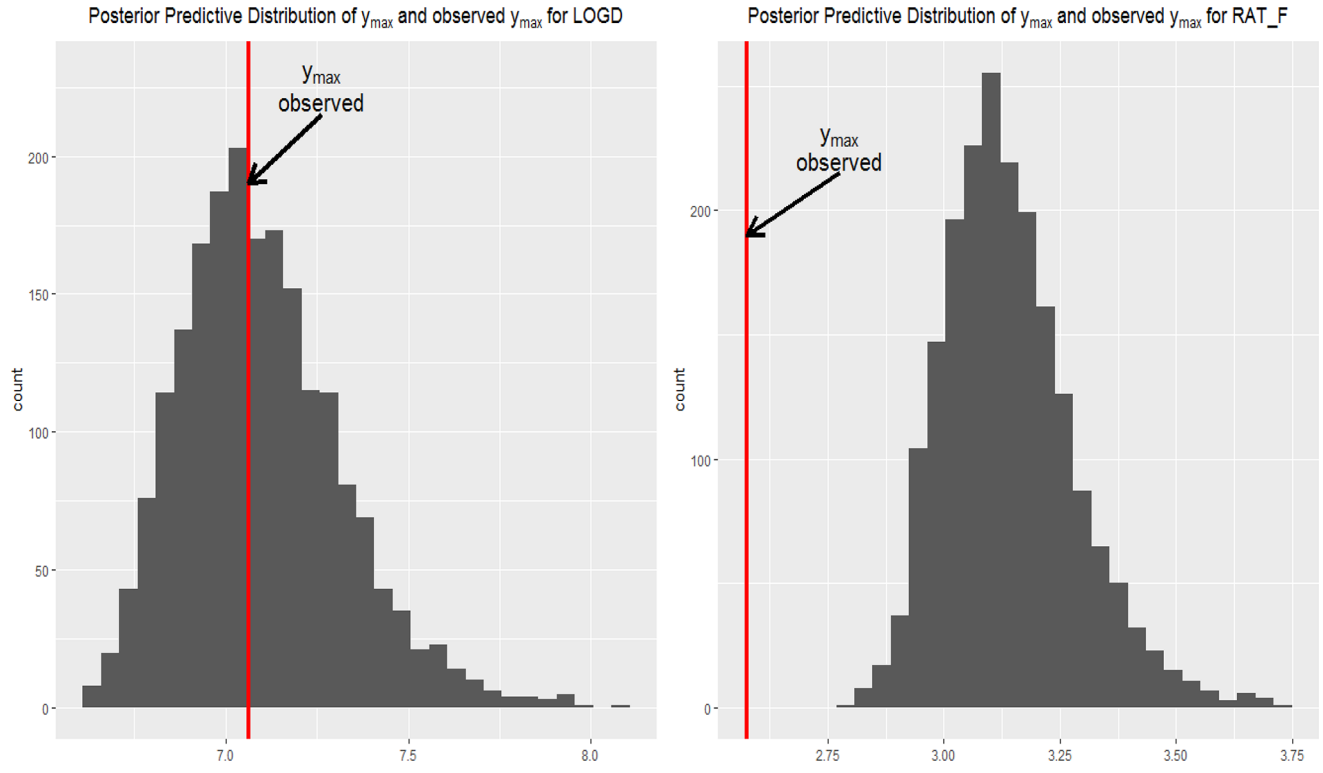



**Figure 14.** Posterior predictive checking.

## CONCLUDING REMARKS

In this paper we studied BART as a model builder for QSAR. BART is formulated entirely in a Bayesian hierarchical modeling framework, which provides great flexibility and power, such as quantifying uncertainty, conducting model diagnosis, and handling truncated data.

Uncertainty is inherent in QSAR. Besides point prediction of an activity, the estimation of the uncertainty of the predicted value is also critical—a higher variation indicates the lack of



DOI: 10.1021/acs.jcim.9b00094
J. Chem. Inf. Model. 2019, 59, 2642−2655

confidence of the prediction. To properly compare BART and RF in terms of quantifying the uncertainty, we proposed a method, unlike previous approaches, for calculating RF prediction intervals that takes into account RF variability due to the random sampling of the training data. We then compared BART with RF and demonstrated that BART provided on average more accurate predicative activity and accurate prediction interval that was not offset by larger width. In this paper, we focused on the uncertainty of RF and BART based approaches. Among the three methods: RF, XGBoost, and BART, XGBoost provided on average the best prediction accuracy. We plan to investigate the uncertainty of other methods, such as XGBoost and Gaussian Process, in the future.

We suggest to provide not only point prediction of molecular activities but corresponding prediction intervals as well to reflect confidence of predictions. Similarly, the variance of variable importance needs to be considered in addition to point estimate when selecting key descriptors.

Instead of being an algorithm, BART is essentially a statistical model. We can evaluate whether the model adequately represents data. Model diagnostic is an essential component of statistical inference. In this study we showed an example of using posterior predictive checking to detect model inadequacy. When using BART, model checking is indispensable since the lack of fit can lead to poor prediction. We need to be cautious when using predictions from a model with inadequacy detected. In this paper, we focus on the BART model; in the Bayesian hierarchical model framework, we can consider different models, either comparing them to choose a better model or implementing model averaging.

Bayesian reasoning combines prior knowledge/experience with current information from data newly collected to conduct statistical analysis. For the truncation of qualified data, we built the knowledge that the true activities are less than the minimum value observed in the model training. In the future we will investigate how to incorporate BART domain knowledge into model building to achieve more accurate prediction and more precise estimation of uncertainty.

## AUTHOR INFORMATION


### Corresponding Authors

*E-mail: dai_feng@merck.com.
*E-mail: vladimir_svetnik@merck.com.

### ORCID

Dai Feng: 0000-0001-7136-5793
Robert P. Sheridan: 0000-0002-6549-1635

### Author Contributions

‖D.F. and V.S. contributed equally to this work


### Notes

The authors declare the following competing financial interest(s): Dai Feng, Vladimir Svetnik and Andy Liaw are employees of MSD (Merck Sharp & Dohme Corp., a subsidiary of Merck & Co. Inc., Kenilworth, NJ, USA). Robert P. Sheridan is an independent contractor paid by MSD. Matthew Pratola is a consultant paid by MSD.

## REFERENCES


(1) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of chemical information and computer sciences* **2003**, *43*, 1947−1958.

(2) Svetnik, V.; Wang, T.; Tong, C.; Liaw, A.; Sheridan, R. P.; Song, Q. Boosting: An Ensemble Learning Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Model.* **2005**, *45*, 786−799.

(3) Sheridan, R. P.; Wang, W. M.; Liaw, A.; Ma, J.; Gifford, E. M. Extreme Gradient Boosting as a Method for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2016**, *56*, 2353−2360.

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

(5) Shafer, G.; Vovk, V. A Tutorial on Conformal Prediction. *Journal of Machine Learning Research* **2008**, *9*, 371−421.

(6) Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R. J.; Wasserman, L. Distribution-free Predictive Inference for Regression. *J. Am. Stat. Assoc.* **2018**, *113*, 1094−1111.

(7) Meinshausen, N. Quantile Regression Forests. *Journal of Machine Learning Research* **2006**, *7*, 983−999.

(8) Wager, S.; Hastie, T.; Efron, B. Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife. *Journal of Machine Learning Research* **2014**, *15*, 1625−1651.

(9) Svensson, F.; Aniceto, N.; Norinder, U.; Cortes-Ciriano, I.; Spjuth, O.; Carlsson, L.; Bender, A. Conformal Regression for Quantitative Structure−Activity Relationship Modeling−Quantifying Prediction Uncertainty. *J. Chem. Inf. Model.* **2018**, *58*, 1132−1140.

(10) Johansson, U.; Boström, H.; Löfström, T.; Linusson, H. Regression Conformal Prediction with Random Forests. *Machine Learning* **2014**, *97*, 155−176.

(11) Burden, F. R. Quantitative Structure-Activity Relationship Studies Using Gaussian Processes. *Journal of chemical information and computer sciences* **2001**, *41*, 830−835.

(12) Cortes-Ciriano, I.; van Westen, G. J.; Lenselink, E. B.; Murrell, D. S.; Bender, A.; Malliavin, T. Proteochemometric Modeling in a Bayesian Framework. *J. Cheminf.* **2014**, *6*, 35.

(13) Obrezanova, O.; Csányi, G.; Gola, J. M.; Segall, M. D. Gaussian Processes: A Method for Automatic QSAR Modeling of ADME Properties. *J. Chem. Inf. Model.* **2007**, *47*, 1847−1857.

(14) Chipman, H. A.; George, E. I.; McCulloch, R. E. BART: Bayesian Additive Regression Trees. *Annals of Applied Statistics* **2010**, *4*, 266−298.

(15) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Model.* **1985**, *25*, 64−73.

(16) Kearsley, S. K.; Sallamack, S.; Fluder, E. M.; Andose, J. D.; Mosley, R. T.; Sheridan, R. P. Chemical Similarity Using Physiochemical Property Descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 118−127.

(17) Breiman, L. Random Forests. *Machine learning* **2001**, *45*, 5−32.

(18) Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of statistics* **2001**, *29*, 1189−1232.

(19) Gelman, A.; Carlin, J. B.; Stern, H. S.; Dunson, D. B.; Vehtari, A.; Rubin, D. B. *Bayesian Data Analysis*; CRC Press: Boca Raton, FL, 2014; Vol. *2*.

(20) Chipman, H. A.; George, E. I.; McCulloch, R. E. Bayesian CART Model Search. *J. Am. Stat. Assoc.* **1998**, *93*, 935−948.

(21) Tierney, L. Markov Chains for Exploring Posterior Distributions. *the Annals of Statistics* **1994**, *22*, 1701−1728.

(22) Wu, Y.; Tjelmeland, H.; West, M. Bayesian CART: Prior Specification and Posterior Simulation. *Journal of Computational and Graphical Statistics* **2007**, *16*, 44−66.

(23) Pratola, M. T. Efficient Metropolis−Hastings proposal mechanisms for Bayesian regression tree models. *Bayesian analysis* **2016**, *11*, 885−911.

(24) Pratola, M. T.; Chipman, H. A.; George, E. I.; McCulloch, R. E. Heteroscedastic BART Using Multiplicative Regression Trees. arXiv preprint arXiv:1709.07542, **2017**.

(25) Nicholls, A. Confidence Limits, Error Bars and Method Comparison in Molecular Modeling. Part 1: the Calculation of Confidence Intervals. *J. Comput.-Aided Mol. Des.* **2014**, *28*, 887−918.




DOI: 10.1021/acs.jcim.9b00094
*J. Chem. Inf. Model.* 2019, 59, 2642−2655

(26) Nicholls, A. Confidence Limits, Error Bars and Method Comparison in Molecular Modeling. Part 2: Comparing Methods. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 103−126.
(27) Wager, S. *randomForestCI: Confidence Intervals for Random Forests*. 2018; R package version 1.0.0.
(28) Meinshausen, N. *QuantregForest: Quantile Regression Forests*. 2017; R package version 1.3-7.



DOI: 10.1021/acs.jcim.9b00094
*J. Chem. Inf. Model.* 2019, 59, 2642−2655