Estimation of Predictive Performance in High-Dimensional Data Settings using Learning Curves

Jeroen M. Goedhart\textsuperscript{a}, Thomas Klausch\textsuperscript{a}, Mark A. van de Wiel\textsuperscript{a}
\textsuperscript{a}Department of Epidemiology and Data Science, Amsterdam University Medical Centers

June 9, 2022

Abstract

In high-dimensional prediction settings, it remains challenging to reliably estimate the test performance. To address this challenge, a novel performance estimation framework is presented. This framework, called Learn2Evaluate, is based on learning curves by fitting a smooth monotone curve depicting test performance as a function of the sample size. Learn2Evaluate has several advantages compared to commonly applied performance estimation methodologies. Firstly, a learning curve offers a graphical overview of a learner. This overview assists in assessing the potential benefit of adding training samples and it provides a more complete comparison between learners than performance estimates at a fixed subsample size. Secondly, a learning curve facilitates in estimating the performance at the total sample size rather than a subsample size. Thirdly, Learn2Evaluate allows the computation of a theoretically justified and useful lower confidence bound. Furthermore, this bound may be tightened by performing a bias correction. The benefits of Learn2Evaluate are illustrated by a simulation study and applications to omics data.

Keywords: High-dimensional data, Omics, Predictive performance, Area under the receiver operating curve, Bootstrap, Cross-validation

\textsuperscript{*}Corresponding author, E-mail address: j.m.goedhart@amsterdamumc.nl
1 Introduction

Nowadays, many learners can predict a response from high-dimensional data. To select the most accurate learner, it is key to reliably estimate the predictive performance with a confidence bound. The predictive performance may be quantified by many different metrics. For classification, examples include the accuracy, the area under the receiver operating curve (AUC) [Hanley and McNeil, 1982], and the brier score [Brier, 1950], whereas for regression one usually reports the mean square error of prediction (PMSE).

It is well-known that performance estimates are optimistically biased when they are based on the same data used to fit the learner. This optimism bias is even more apparent for high-dimensional learners, as such learners overfit quickly. Performance estimates should therefore be based on independent test data. However, for many high-dimensional prediction studies, a test set is unavailable. Therefore, learning the model and estimating the performance should be based on a single, often small-sized, data set. In this paper, we present a novel methodology to estimate the predictive performance in such a scenario.

The standard method for performance estimation in small sample size settings is resampling such as repeated hold-out [Burman, 1989], K-fold cross-validation [Stone, 1974], or bootstrapping [Efron and Tibshirani, 1994]. These techniques repeatedly partition or resample the available data in a training set of a fixed size, to fit the model, and a test set, to estimate the performance. An acknowledged challenge for these techniques is to choose the size of the training sets. Training sets that are close in size to the complete data set lead to an almost unbiased predictive performance estimate. However, such training sets also induce a large variability of the performance estimate because the test sets are small. Resampling techniques are, therefore, faced with a bias-variance trade-off, which leads to conservative and often useless confidence bounds for the performance. Advances in these techniques improve the uncertainty estimates [Michiels et al., 2005, Jiang et al., 2008], but neither provide a theoretical justification nor a point estimate.

Another performance estimation strategy is to combine resampling with an asymptotic or parametric method to construct a confidence bound. For the cross-validated AUC, a confidence bound based on influence functions was derived [LeDell et al., 2015], and Monte Carlo simulation was combined with a parametric model to construct a confidence bound for the cross-validated accuracy [Dobbin, 2009]. However, the former bound is too liberal for small sample sizes and the latter method only applies to linear learners.
Moreover, both methods lack generality because they only apply to a specific predictive performance metric.

To address the aforementioned issues, we present Learn2Evaluate, a learning-curve-based performance estimation framework. A learning curve combines resampling with smoothing by fitting a monotone curve depicting the predictive performance on left-out samples as a function of the size of the training set [Cortes et al., 1993, Mukherjee et al., 2003]. The learning curve enables us to obtain a point estimate of the performance at the full sample size and to determine an optimal training set size to construct a conservative confidence bound. Additionally, this bound can be tightened by performing a bias correction. The idea of finding an optimal training set size builds on previous work by Dobbin and Simon [2011]. They employed learning curves to find a good training set size in single split sample approach. In contrast, we consider multiple splits into a training and test set, and we focus on a confidence bound.

Learn2Evaluate has several advantages compared to other performance estimation techniques. Firstly, Learn2Evaluate is generic and applies to any learner and any performance metric. Secondly, it renders a point estimate and a confidence bound. Importantly, this confidence bound is theoretically justified, as we will prove type-I error control, and it explicitly takes the size of the training set into account.

In addition, because Learn2Evaluate is based on learning curves, it automatically offers a graphical technique to study several aspects of a learner. Firstly, the slope of the curve at the total sample size allows a qualitative assessment of the benefit of future samples. Secondly, learning curves provide a more complete comparison between different learners than just performance point estimates at a fixed training set size. Lastly, one may observe that prediction models learn at different rates, which could lead to a deeper understanding of learning behaviors.

The remainder of this paper is organized as follows. In Section 2, we describe Learn2Evaluate and provide a theoretical justification for the confidence bound. We present simulation results in Section 3 and in Section 4, we illustrate the benefits of Learn2Evaluate on omics data. We end with some conclusive remarks in Section 5.
2 Methods

2.1 Target parameter
Let our data $D = \{(X_i, Y_i)\}_{i=1}^{N}$ consist of $N$ observations of a $p$-dimensional feature vector $X_i$ with a corresponding response variable $Y_i$. The goal in prediction problems is to fit a learner $\varphi_D$ on data $D$ that reliably predicts $Y$ from $X$. After learner $\varphi_D$ is fitted, we wish to evaluate the predictive performance $\rho_D$, i.e. the quality of predictions of $\varphi_D$ of yet unseen data points. Subscript $D$ indicates that $\rho_D$ is conditional on the data $D$.

2.2 Predictive performance estimation
To evaluate $\rho_D$, we report a point estimate $\hat{\rho}_D$ and a lower confidence bound $L_D(\alpha)$ for $\rho_D$ that satisfies $P(L_D(\alpha) \leq \rho_D) \geq 1 - \alpha$. We consider the case where a large test data set is not available. Learning $\varphi_D$ and estimating $\hat{\rho}_D$ and $L_D(\alpha)$ should therefore be solely based on data $D$.

To use $D$ efficiently, one may apply resampling techniques. These techniques create multiple training sets, on which $\varphi$ is learned, and complementary test sets, on which the estimates $\hat{\rho}_D$ and $L_D(\alpha)$ are based. Since $\varphi$ is now learned on a subset of $D$, it is more difficult to find the structure in $X$ that explains $Y$. This often leads to a pessimistically biased estimate of $\rho_D$. Larger training sets reduce this bias, but also lead to smaller test sets, which in turn increases the variance of the performance estimate. Resampling-based estimators are therefore faced with a bias-variance trade-off. This trade-off makes it particularly difficult to obtain a practical confidence bound as such a bound depends on both the bias and the variance.

To address the bias-variance trade-off, we present a novel learning-curve-based performance estimation method. In the following, we formally define the learning curve and discuss how we use it for predictive performance estimation. Here, we distinguish between increasing performance metrics such as the AUC, which are metrics that increase when more signal is captured, and decreasing performance metrics such as the PMSE. Our method is presented for the AUC because classification is a frequent application in high-dimensional and small sample size settings. Extending our method to decreasing performance metrics is straightforward and is explained for the PMSE. Our method, which we call Learn2Evaluate, is summarized in Figure 1.
2.3 Definition of the learning curve

A learning curve is constructed by first applying repeated hold-out estimation for several training set sizes, followed by a smoothing step.

2.3.1 Repeated hold-out estimation

To define repeated hold-out estimation, we first introduce single hold-out estimation. Let $s^k_n$ be the $k$th realization of a random subset of $D$ with size $n \leq N$. This subset is taken without replacement and is balanced with respect to the response if $Y$ is binary. For continuous $Y$, we do not impose the balancing restriction. We then define $\rho_{s^k_n}$ as the predictive performance when $s^k_n$ is used to build a learner $\varphi_{s^k_n}$. We estimate $\rho_{s^k_n}$ based on the complement or hold-out set $D \setminus s^k_n$. This yields the single hold-out performance estimate $\hat{\rho}_{s^k_n}$ and a lower confidence bound $L_{s^k_n}(\alpha)$. We assume that a method to construct $L_{s^k_n}(\alpha)$ exists. For the AUC, we employ the nonparametric method of DeLong et al. [1988] implemented in the R package pROC [Robin et al., 2011]. For decreasing performance metrics, such as the PMSE, we estimate an upper confidence bound.

The single hold-out estimate depends on the subset $s^k_n$. To reduce this dependency, we apply repeated hold-out estimation. We define $\rho_n$ as the average performance over all $V$ possible subsets of $D$ of size $n \leq N$,

$$\rho_n = \frac{1}{V} \sum_{k=1}^{V} \rho_{s^k_n},$$

with $V = \binom{N}{n}$ for unrestricted sampling, and $V = \binom{N_+}{n_+}\binom{N_-}{n_-}$ for balanced sampling, with subscript $+/-$ denoting the number of positive/negative cases of the given (sub)sample size. The average performance is estimated by the repeated hold-out estimate $\hat{\rho}_n$, which is based on $K \leq V$ subsets,

$$\hat{\rho}_n = \frac{1}{K} \sum_{k=1}^{K} \hat{\rho}_{s^k_n}. \quad (1)$$

Repeated hold-out estimation also yields $K$ lower confidence bounds $L_{s^k_n}(\alpha)$. In Theorem 1, we prove how to aggregate these bounds to obtain a $(1 - \alpha)$ confidence bound for $\rho_D$.

Conventional repeated hold-out estimation uses (1) as a point estimate for the performance $\rho_D$. A learning curve, however, relies on first evaluating (1) for a sequence of $J$ subsample sizes $n_1 < \ldots < n_J$, yielding a learning trajectory (Figure 1, black dots),

$$\{(n_j, \hat{\rho}_{n_j})\}_{j=1}^{J}. \quad (2)$$
The learning curve is then obtained by smoothing this trajectory and the fit provides an estimate of $\rho_D$.

### 2.3.2 Smoothing

A learning curve $f(n)$ (Figure 1, solid purple line) is a fit of the learning trajectory, i.e. (2), rendering a relationship between the subsample size and the predictive performance. We assume that this relationship is monotonically increasing,

$$\forall j > 0 : \rho_{n_j} \leq \rho_{n_{j+1}}.$$  \hspace{1cm} (3)

Assumption (3) is a natural one: the average performance of a learner trained on larger subsets is likely higher than that of the same learner trained on smaller subsets.

In addition, we assume that the learning curve is concave after some threshold subsample size $n_{tr}$, $\forall n_j \geq n_{tr}, j > 0 : (\rho_{n_{j+2}} - \rho_{n_{j+1}}) / (n_{j+2} - n_{j+1}) \leq (\rho_{n_{j+1}} - \rho_{n_j}) / (n_{j+1} - n_j)$. This assumption is employed to stabilize the curve. In some situations, the concavity assumption may not hold for the considered trajectory of subsample sizes, i.e. $n_1 \leq n_{tr}$. This can be assessed visually from the learning trajectory.

For example, in some situations, lasso regression is known to exhibit a phase transition [Donoho and Tanner, 2005], which would lead to a S-shaped learning curve. For such an instance, we propose to first determine $n_{tr}$, i.e. the inflection point of the S-shaped curve, followed by fitting a learning curve with smallest subsample size $n_1 = n_{tr}$. In supplementary Section 1, we show a method to determine $n_{tr}$. Note that the case $n_J \leq n_{tr}$ is a strong indication for acquiring more samples before developing a learner.

For decreasing performance metrics, the assumptions should be reversed, leading to a monotonically decreasing and convex curve.

To fit the learning curve, we consider a parametric model, based on an inverse power law, and a nonparametric model, based on a constrained regression spline. Use of power laws for fitting learning curves has been empirically established [Cortes et al., 1993, Mukherjee et al., 2003, Hess and Wei, 2010]. We examine constrained regression splines because they do not rely on a specific shape of $f$, while the constraints prevent a too flexible fit.
Inverse power law

To fit an inverse power law, we perform nonlinear least squares regression, with

\[ f(n) = \delta - \beta n^{-\gamma}. \]  

(4)

This function grows, with increasing subsample size \( n \), asymptotically to parameter \( \delta \), with \( 0.5 \leq \delta \leq 1 \) for the AUC. The learning rate \( \beta \) and the decay rate \( \gamma \) are positive [Cortes et al., 1993]. Derivative inspection reveals that (4) satisfies the monotony and concavity assumptions. To estimate the parameters \( \delta, \beta, \) and \( \gamma \), we solve

\[
\left( \hat{\delta}, \hat{\beta}, \hat{\gamma} \right) = \arg\min_{\delta,\beta,\gamma} \frac{1}{J} \sum_{j=1}^{J} \left( \delta - \beta n_j^{-\gamma} - \hat{\rho}_n \right)^2, \text{ s.t. } 0.5 \leq \delta \leq 1, \ 0 \leq \beta, \text{ and } 0 \leq \gamma,
\]  

(5)

by using the L-BFGS-B method [Byrd et al., 1995] of the base R optim function. For decreasing metrics, we have \( f(n) = \delta + \beta n^{-\gamma} \), with \( 0 \leq \delta, \ 0 \leq \beta, \) and \( 0 \leq \gamma \).

Constrained regression spline

To fit a constrained regression spline, we employ an algorithm proposed by Ng and Maechler [2007] implemented in the R package COBS [Ng and Maechler, 2020]. Briefly, the function \( f \) is approximated by fitting piecewise polynomial basis functions over different regions of the subsample size \( n \). These regions are specified by \( M \) knots, which leads to a B-spline basis \( \pi(n) \in R^{M+\kappa} \), \( \pi(n) = (\pi_1(n), \ldots, \pi_{M+\kappa}(n))^T \), where \( \kappa \) is the maximum degree of the polynomial. Then, we define the learning curve as

\[ f(n) = \pi(n)^T \theta, \]  

(6)

where \( \theta \in R^{M+\kappa} \), a vector of regression parameters, is estimated by solving

\[
\hat{\theta} = \arg\min_{\theta} \sum_{j=1}^{J} | \hat{\rho}_n - \pi(n_j)^T \theta |, \text{ subject to }
\]  

(7)

\[
\pi(n_j)^T \theta \leq \pi(n_{j+1})^T \theta \quad \text{and} \quad \frac{\pi(n_{j+1})^T \theta - \pi(n_j)^T \theta}{(n_{j+1} - n_j)} \leq \frac{\pi(n_j)^T \theta - \pi(n_{j-1})^T \theta}{(n_j - n_{j-1})}.
\]

Again, the constraints should be decreasing and convex for decreasing performance metrics.

2.4 Learning-curve-based performance estimates

Once the learning curve is fitted, we use it to obtain a point estimate and a confidence bound of the target performance \( \rho_D \).
2.4.1 Point estimate

For the point estimate, we extrapolate the learning curve to the full sample size \(N\), i.e. we estimate \(\rho_D\) by \(f(N)\) (Figure 1, black cross). Extrapolation to \(N\) circumvents the pessimistic bias of resampling techniques, induced by learning the model on a subset of \(D\). Additionally, smoothing of the learning trajectory, combined with the constraints, may denoise the individual repeated hold-out estimates.

2.4.2 Lower confidence bound

For the confidence bound, we employ the learning curve to find a good training set size that takes the bias-variance trade-off into account.

We first prove a \((1 - \alpha)\) lower confidence bound for parameter \(\rho_D\). We start by posing the following assumption:

\[
\frac{1}{V} \sum_{k=1}^{V} P(L_{sk_n} > \rho_D) \leq \frac{1}{V} \sum_{k=1}^{V} P(L_{sk_n} > \rho_{sk_n})
\]  

(8)

Note that this assumption aligns well with the monotony assumption (3): for \(\rho_D\) being larger than \(\rho_{sk_n}\) on average it is reasonable to assume that the average exceedance probability for threshold \(\rho_D\) is smaller than that for \(\rho_{sk_n}\).

**Theorem 1.** Let \(L_{sk_n}(\alpha)\) be the \((1 - \alpha)\) lower confidence bound for \(\rho_{sk_n}\), and let \(S^k_n\) be a random subset of size \(n\), with corresponding random lower bound \(L_{S^k_n}(\alpha)\). Assume (8) and suppose there exists a monotone transformation \(\xi\) of \(L_{S^k_n}\) such that \((\xi(L_{S^k_n}))_{k=1}^{K}\) follows a multivariate normal distribution with standard normal marginals, then

\[
\forall n \leq N : P(L_n(\alpha) \leq \rho_D) \geq 1 - \alpha,
\]

where \(L_n(\alpha) = \text{med}(L_{S^k_n}(\alpha))_{k=1}^{K}\).

**Proof.** First note that from (8) we have

\[
P(L_{S^k_n} \leq \rho_D) = \sum_{k=1}^{V} P(S^k_n = s^k_n)P(L_{S^k_n} \leq \rho_D|S^k_n = s^k_n)
\]

\[
= \frac{1}{V} \sum_{k=1}^{V} P(L_{sk_n} \leq \rho_D) \geq \frac{1}{V} \sum_{k=1}^{V} P(L_{sk_n} \leq \rho_{sk_n}) = 1 - \alpha,
\]  

(9)
as $L_{sk}^n$ is a $(1 - \alpha)$ lower confidence bound for $\rho_{sk}$.

Next, we have

$$P\left( L_n(\alpha) \leq \rho_D \right) = P \left( \text{med} \left( L_{sk}^n(\alpha) \right)_{k=1}^K \leq \rho_D \right)$$

$$= P \left( \xi \left( \text{med} \left( L_{sk}^n(\alpha) \right)_{k=1}^K \right) \leq \xi(\rho_D) \right)$$

$$= P \left( \text{med} \left( \xi(L_{sk}^n(\alpha)) \right)_{k=1}^K \leq \xi(\rho_D) \right).$$

Then, (9) and $\xi$ being monotone imply $P\left( \xi(L_{sk}^n(\alpha)) \leq \xi(\rho_D) \right) \geq 1 - \alpha$. Hence $\xi(\rho_D) \geq \Phi^{-1}(1 - \alpha)$, with $\Phi$ the univariate standard normal c.d.f., as $\xi(L_{sk}^n(\alpha)) \sim N(0,1)$. Next, we follow Th.1 in [van de Wiel et al., 2009], which states that for multivariate normal and marginally standard normal random variables $Z_k$: $P \left( \text{med}(Z_k)_{k=1}^K \leq x \right) \geq \Phi(x)$. Finally, substitute $x = \xi(\rho_D)$:

$$P \left( \text{med} \left( \xi(L_{sk}^n(\alpha)) \right)_{k=1}^K \leq \xi(\rho_D) \right) \geq \Phi(\xi(\rho_D)) \geq \Phi \left( \Phi^{-1}(1 - \alpha) \right) = 1 - \alpha,$$

which, in combination with (10), completes the proof.

Note that the strengths of Theorem 1 are that 1) it only requires the existence of $\xi$; we do not need to know what it actually is; and 2) for any continuous random variable $Z$ with finite mean there always exist a monotone $\xi$ such that marginally $\xi(Z) \sim N(0,1)$. Because a multivariate normal distribution can always be decomposed as the sum of the marginals and the copula [Sklar, 1959], the multivariate assumption in Theorem 1 is effectively only an assumption on the dependencies between copies of $\xi(L_{sk}^n)$.

Theorem 1 also holds for decreasing performance metrics and an upper confidence bound.

Determining a good subsample size

Theorem 1 implies that we can determine a lower confidence bound $L_n(\alpha)$ for $\rho_D$ at any subsample size $n \leq N$. To determine an optimal $n$, we explicitly take the bias-variance trade-off into account by employing the saturation level of the learning curve. When a learner has converged at subsample size $n \ll N$, relatively many samples can be used for testing because this will hardly impact the (empirical) bias, i.e. $\hat{\text{bias}}(n) = f(N) - f(n)$. The learning curve enables us to find a $n$, which we call $n_{opt}$, such that the bias is not too large, and the test set is not too small, which would lead to high variance. To determine $n_{opt}$, we minimize the mean square error, i.e.

$$n_{opt} = \arg \min_n \left\{ \text{bias}^2(n) + \text{Var}(n) \right\},$$

(11)
with \( \hat{\text{bias}}(n) \) defined above. For \( \hat{\text{Var}}(n) \), we employ an asymptotic variance estimator for the given performance metric. For the AUC, we employ a formula derived by Bamber [1975], and for the PMSE, Faber [1999] derived an estimator. These estimators are found in supplementary Section 2.

If no asymptotic variance estimator is available, we suggest to control for the empirical bias. This method is described in supplementary Section 3.

![Figure 1: Summary of Learn2Evaluate. First, we estimate the AUC for several training sample sizes \( n_j \) by repeated hold-out estimation, rendering a learning trajectory (black dots). We smooth this trajectory by an inverse power law or a constrained regression spline, leading to a relationship \( f(n) \) between the training set size \( n \) and the AUC. We then estimate the predictive performance \( \rho_D \) by \( f(N) \) (black cross). The learning curve also allows us to determine a lower confidence bound (upward triangle) at an optimal training set size \( n_{\text{opt}} \) (dotted vertical line), which we determine by MSE minimization. To obtain a tighter confidence bound, a bias correction is performed (downward triangle).](image)

### 2.4.3 Bias-corrected confidence bound

Our predictive performance point estimate for \( \rho_D \) is \( f(N) \) (Figure 1, black cross) and our confidence bound for \( \rho_D \) equals \( L_{n_{\text{opt}}} (\alpha) \) (Figure 1, upward triangle), with \( n_{\text{opt}} \) determined by (11). This confidence bound may be tightened by performing a bias correction. \( L_{n_{\text{opt}}} \) is in principle a confidence bound for \( \rho_{n_{\text{opt}}} \), although it is also valid for \( \rho_D \). Hence, we may perform an empirical bias correction to \( L_{n_{\text{opt}}} \) (Figure 1, downward triangle), which
leads to

\[ L_{bc}^{\text{opt}}(\alpha) = L_{\text{opt}}(\alpha) + (f(N) - f(n_{\text{opt}})). \]  \hfill (12)

Correcting for the empirical bias depends on the validity of our point estimate \( f(N) \), which we assess in a simulation study.

3 Simulation study

To verify Learn2Evaluate, we evaluate the quality of the point estimate \( f(N) \), and the theoretically justified confidence bound \( L_{\text{opt}}(\alpha) \) and its bias-corrected version \( L_{bc}^{\text{opt}}(\alpha) \) in a high-dimensional simulation study. We focus on classification and the AUC as performance metric.

3.1 Description of experiments

The binary response is generated via

\[ Y_i \sim \text{Bern} \left( \frac{\exp(X_i \beta)}{1 + \exp(X_i \beta)} \right), \quad i = 1, \ldots, N, \]  \hfill (13)

with covariates \( X_i \sim N(0_p, \Sigma_{p \times p}) \), where \( p = 2000 \), and elements of the parameter vector \( \beta_1 \ldots \beta_p \sim \text{Expo}(\nu) \). To mimic a realistic correlation structure between the features, we define \( \Sigma_{p \times p} \) as the estimated variance-covariance matrix from a real high-dimensional omics data set described in Best et al. [2015], from which we randomly select 2000 genes as covariates. This matrix is estimated by a shrinkage method [Schäfer and Strimmer, 2005] implemented in the R package corpcor [Schäfer et al., 2017]. Parameter vector \( \beta \) is defined such that a dense covariate-response structure is present, which is often the case in omics applications [Boyle et al., 2017]. We vary the sample size \( N \) (\( N = 100 \) and \( N = 200 \)) and the signal in the data by tuning rate parameter \( \nu \) (\( \nu = 1000 \) and \( \nu = 100 \)).

For each setting, specified by \( N \) and \( \nu \), we simulate \( R = 1000 \) data sets. We then apply Learn2Evaluate to each data set for three learners: ridge regression, lasso regression, both implemented in the R package glmnet [Friedman et al., 2010], and random forest, for which we employ the R package randomforestSRC [Ishwaran and Kogalur, 2021]. We generate learning curves by first estimating the AUC by repeated hold-out estimation (50 repeats) for ten subsample sizes \( n_j \) homogeneously spread over the sample size range \([20, N - 10]\). For ridge and lasso regression, we estimate the penalty parameter for each \( n_j \)
by the median of repeated (5 times) 10-fold CV. This median is fixed for the 50 repeats of each \( n_j \). We obtain the learning curve by smoothing the AUC point estimates at different subsample sizes by either the inverse power law or the constrained regression spline. We then estimate the AUC by \( f(N) \) and 95% lower confidence bounds by \( L_{n_{opt}}(0.05) \) and its bias-corrected version \( L_{n_{opt}}^{bc}(0.05) \). We determine \( n_{opt} \) by MSE minimization given by (11), using the asymptotic AUC variance estimator (eq. 5 of supplementary Section 2).

We denote \( f(N) \) by \( f_r(N) \) for the \( r \)th simulated data set, and we approximate the true AUC by \( \text{AUC}_r \): the AUC of the \( r \)th learned model evaluated on 25,000 independent test samples. The average \( \text{AUC}_r \) (of \( R = 1000 \) simulation runs) is given in the Appendix (Table A1).

We assess the quality of point estimates \( f_r(N) \) by the root mean square error (RMSE) and the averaged bias (Bias),

\[
\text{RMSE} = \sqrt{\frac{1}{R} \sum_{r=1}^{R} (f_r(N) - \text{AUC}_r)^2} \quad \text{and} \quad \text{Bias} = \frac{1}{R} \sum_{r=1}^{R} (f_r(N) - \text{AUC}_r). \tag{14}
\]

Confidence bounds \( L_{n_{opt}}(0.05) \) and \( L_{n_{opt}}^{bc}(0.05) \) are evaluated by their coverage, i.e. the proportion of times the confidence bound is lower than \( \text{AUC}_r \), and the distance of each confidence bound to its corresponding \( \text{AUC}_r \).

We compare Learn2Evaluate, fitted by an inverse power law (L2E-P) or a spline (L2E-S), with two methods that produce a point estimate and a lower confidence bound. Firstly, we consider ten-fold cross-validation (10F-CV). Point estimates are given by the average of the ten test folds and a lower confidence bound is obtained by employing an asymptotic variance estimator, which is derived by LeDell et al. [2015] and implemented in the R package cvAUC [LeDell et al., 2014]. Secondly, we consider leave-one-out bootstrapping (LOOB) [Efron and Tibshirani, 1994] with 500 bootstrapped training sets and complementary test sets. From the 500 AUC estimates, we obtain a point estimate and a lower confidence bound by taking the average and the 5% quantile, respectively. Details on the code are found in supplementary Section 9.

### 3.2 Point estimate evaluation

The RMSE and the averaged bias of the point estimates are given in Table 1. The inverse power law (L2E-P) is the overall winner in terms of RMSE, although sometimes, in
Table 1: Root mean square error (RMSE) and averaged bias (Bias) of Learn2Evaluate, fitted by an inverse power law (L2E-P) or a constrained regression spline (L2E-S), 10-fold CV (10F-CV), and leave-one-out bootstrapping (LOOB).

|          | Ridge          | Lasso          | RF            |
|----------|----------------|----------------|---------------|
|          | RMSE | Bias | RMSE | Bias | RMSE | Bias |
| L2E-P    | 0.053 | 0.004 | 0.058 | 0.009 | 0.048 | 0.020 |
| N = 100, |      |      |      |      |      |      |
| L2E-S    | 0.054 | 0.007 | 0.070 | 0.025 | 0.050 | 0.026 |
| ν = 1000 | 10F-CV | -0.001 | 0.084 | -0.009 | 0.051 | 0.017 |
| LOOB     | 0.071 | -0.039 | 0.057 | -0.020 | 0.050 | -0.013 |
| L2E-P    | 0.037 | 0.002 | 0.048 | 0.002 | 0.037 | 0.002 |
| N = 100, |      |      |      |      |      |      |
| L2E-S    | 0.038 | 0.004 | 0.056 | 0.009 | 0.038 | 0.005 |
| ν = 1000 | 10F-CV | 0 | 0.062 | -0.005 | 0.039 | 0.001 |
| LOOB     | 0.048 | -0.028 | 0.051 | -0.025 | 0.043 | -0.018 |
| L2E-P    | 0.036 | 0.001 | 0.041 | 0.004 | 0.034 | 0.002 |
| N = 200, |      |      |      |      |      |      |
| L2E-S    | 0.038 | 0.004 | 0.046 | 0.010 | 0.034 | 0.004 |
| ν = 1000 | 10F-CV | -0.002 | 0.050 | -0.004 | 0.035 | 0.009 |
| LOOB     | 0.060 | -0.041 | 0.052 | -0.034 | 0.036 | -0.009 |
| L2E-P    | 0.024 | 0 | 0.031 | 0.001 | 0.026 | 0.007 |
| N = 200, |      |      |      |      |      |      |
| L2E-S    | 0.025 | 0.002 | 0.035 | 0.004 | 0.027 | 0.008 |
| ν = 100  | 10F-CV | -0.001 | 0.036 | -0.005 | 0.026 | 0.008 |
| LOOB     | 0.035 | -0.022 | 0.037 | -0.022 | 0.026 | -0.006 |

For the averaged bias, the situation is less straightforward. Learn2Evaluate always has a positive bias, whereas the competing methods mostly have a negative bias. The bootstrap often has the most biased estimate. Only for the lasso and the random forest in the N = 100, ν = 1000 setting and the random forest in the N = 200, ν = 100 setting is this not the case. L2E-P, which is always (somewhat) less biased than L2E-S, is competitive to 10-fold CV (10F-CV).
A comparison between the learners reveals that the lasso has the largest RMSE for all simulation settings and predictive performance estimators, except for the bootstrap in settings \( N = 100, \nu = 1000 \) and \( N = 200, \nu = 1000 \). This finding may be explained by the slope of the learning curve at the end of the learning trajectory. A large slope causes a drop in predictive performance when it is estimated on a subset of the data set, as is the case for 10-fold CV and bootstrapping. Learn2Evaluate also suffers from a larger slope because extrapolation to \( N \) becomes more sensitive to errors.

### 3.3 Confidence bound evaluation

Coverage probabilities for confidence bounds \( L_{n_{opt}}(0.05) \) and \( L_{n_{opt}}^{bc}(0.05) \) are given in Table 2, with \( n_{opt} \) determined by MSE minimization. We also evaluate the asymptotic 10-fold CV confidence bound estimator for the AUC (Le Dell) and leave-one-out bootstrapping (LOOB). We show results for the learning curve fitted by an inverse power law. In supplementary Section 3 (Table S1), we show coverage results when \( n_{opt} \) is determined by controlling for the empirical bias instead of MSE minimization. These coverage results are similar as in Table 2. Results for the constrained regression spline are found in supplementary Section 4 (Table S2).

Table 2: Coverage results for the 95% lower confidence bounds of Learn2Evaluate with \( L_{n_{opt}}^{bc} \) and without bias correction \( L_{n_{opt}} \), the asymptotic AUC variance estimator of Le Dell (Le Dell), and leave-one-out bootstrapping (LOOB). The learning curve is fitted by a power law and \( n_{opt} \) is determined by MSE minimization.

| \( N \) | \( \nu \) | Ridge | Lasso | RF | Ridge | Lasso | RF |
|-------|-------|-------|-------|-----|-------|-------|-----|
| \( N = 100, \nu = 1000 \) | Le Dell | 0.949 | 0.935 | 0.937 | \( L_{n_{opt}}^{bc} \) | 0.949 | 0.935 | 0.937 |
| | LOOB | 0.895 | 0.851 | 0.876 | \( L_{n_{opt}} \) | 0.900 | 0.960 | 0.995 |
| \( N = 200, \nu = 1000 \) | Le Dell | 0.951 | 0.969 | 0.970 | \( L_{n_{opt}}^{bc} \) | 0.961 | 0.969 | 0.970 |
| | LOOB | 0.911 | 0.874 | 0.884 | \( L_{n_{opt}} \) | 0.996 | 1.000 | 0.992 |

Table 2 demonstrates that \( L_{n_{opt}}(0.05) \) controls the type-I error well below the nominal
level, while being less conservative than leave-one-out bootstrapping. Coverage is closer to nominal level for $L_{n_{opt}}^{bc}(0.05)$. In one instance (random forest, $N = 200$, $\nu = 100$, $L_{n_{opt}}^{bc}$), the coverage drops just below 95% when taking the binomial error of the coverage estimate into account. When the learning curve is fitted by a constrained regression spline, $L_{n_{opt}}^{bc}(0.05)$ is too liberal in three cases (supplementary Table S2). This result is explained by the worse point estimates of the spline compared to the power law. The asymptotic confidence bound estimator (Le Dell) is too liberal for small sample sizes, similar as in LeDell et al. [2015].

In supplementary Section 5 (Figures S2, S3, S4, and S5), we depict boxplots of the distance of the lower confidence bound to the corresponding target parameter $AUC_r$ for the bootstrap, and Learn2Evaluate with ($L_{n_{opt}}^{bc}$) and without ($L_{n_{opt}}$) bias correction. These boxplots show that $L_{n_{opt}}$ and $L_{n_{opt}}^{bc}$ are on average closer to the target parameter $AUC_r$ than bootstrapping, with an exception of $L_{n_{opt}}$ for the lasso. The bias correction shortens the distance compared to the confidence bound without bias correction.

The coverage results and the boxplots also illustrate that $L_{n_{opt}}$ for lasso regression is further away from $AUC_r$ than for ridge regression and random forest. The average optimal training set size $n_{opt}$ for lasso regression is larger than for ridge regression and random forest (Appendix Table B1). Hence, the average test set size for lasso regression is smaller, which leads to wider confidence bounds. For $L_{n_{opt}}^{bc}$, the difference between the learners is smaller.

4 Applications

4.1 Classification

We apply Learn2Evaluate to messenger-RNA sequencing data, extracted from blood platelets, as described in Best et al. [2015]. These experiments demonstrated that RNA profiles of blood platelets are a promising diagnostic tool for early cancer detection.

RNA profiles of blood platelets from 230 patients, having one of the in total six tumor types, and 55 healthy controls were obtained. The raw data are online available in the GEO database (GEO: GSE68086). Data preprocessing is described in Novianti et al. [2017]. Processed data are available via https://github.com/JeroenGoedhart/Learn2Evaluate as well as the R code to apply Learn2Evaluate to these data. For
additional details on the data and the code, we refer to supplementary Section 9.

Here, we consider three binary classification cases: non-small-cell lung cancer (NSCLC, \( n = 60 \)) versus the control group (\( n = 55 \)) using \( p = 19,300 \) transcripts (Figure 2a), breast cancer (Breast, \( n = 40 \)) versus NSCLC using \( p = 18,741 \) transcripts (Figure 2b), and breast cancer versus pancreas cancer (Pancreas, \( n = 35 \)) using \( p = 18,350 \) transcripts (Figure 2c). For three other binary classification experiments, we refer to supplementary Section 6 (Figure S6).

Next, we generate learning curves for ridge regression (black), lasso regression (blue), and random forest (green). Learning curves are obtained by the same procedure as in the simulation study. For ten homogeneously spaced subsample sizes, we estimate the predictive performance, quantified by the AUC, by repeated (50 times) hold-out estimation. This renders the learning trajectory (dots), i.e. (2). The maximum subsample size equals \( N - 10 \). We varied the number of different subsamples and repeats to establish that these defaults yield stable estimates (supplementary Section 8, Table S3). We then fit an inverse power law given by (4) (solid line) because this fit obtained the best performance estimates in the simulation.

We construct confidence bounds based on Learn2Evaluate without bias correction (\( L_{n_{opt}} \), upward triangle), Learn2Evaluate with bias correction (\( L_{n_{opt}}^{bc} \), downward triangle), and leave-one-out bootstrapping (Bootstrap, square). We use MSE minimization, i.e. (11), to determine an optimal training subsample size \( n_{opt} \) (dashed vertical line). We also display the point estimates of Learn2Evaluate (star). Figure 2 and supplementary Figure S6 illustrate that the theoretically justified confidence bound of Learn2Evaluate (\( L_{n_{opt}} \)) is, in most cases, slightly tighter than the bootstrap confidence bound, which agrees with the simulation results. A bias correction (\( L_{n_{opt}}^{bc} \)) tightens the theoretically justified bound as expected. Furthermore, Figure 2 shows that \( n_{opt} \) adapts to the saturation level of the learning curve. A more saturated curve leads to a smaller training set and hence to a larger test set to determine the lower bound.

Figure 2 and supplementary Figure S6 also illustrate the additional benefits of using learning curves for predictive performance estimation. Firstly, the curves indicate how future samples increase the AUC. For lasso regression, which has in most cases a large slope, additional samples are expected to increase the predictive performance. The ridge and random forest level off completely in Figure 2a, suggesting that future samples do
Figure 2: Learning curves for three classification tasks. In each task, we produce a learning curve for a ridge (black), a lasso (blue), and a random forest (green) model. For each learner, we depict the learning trajectory (dots) together with the inverse power law fit (solid line). We show confidence bounds of Learn2Evaluate with ($L_{nc}^{opt}$, downward triangle) and without ($L_{n}^{opt}$, upward triangle) bias correction, and bootstrapping (Bootstrap, square). We use MSE minimization to determine $n_{opt}$ (dashed vertical line).

Secondly, the curves provide a more complete comparison between learners than just employing single point estimates. For example, point estimates of the lasso and the ridge are almost similar in Figure 2b. The lasso model may still be preferred because its learning curve has a larger slope, which likely leads to a larger increase in the AUC when additional training samples are available.

Finally, the curves show different learning rates. The lasso model tends to be a slow

not increase the AUC.
learner. At small subsample sizes, it performs much worse than the ridge and random forest model, but it catches up at larger subsample sizes.

### 4.2 Regression

The level of methylation of CpG islands has been shown to be predictive of age [Numata et al., 2012]. Here, we apply Learn2Evaluate to a DNA methylation data set ($N = 108$ and $p = 2289$), which is used to accurately predict age. The predictive performance is quantified by the PMSE. Details on this data set and the learning curves are found in supplementary Section 7 (Figure S7).

Figure S7 illustrates that the decreasing and convex power law fits the learning trajectories for the PMSE well. The upper confidence bounds of Learn2Evaluate are in most cases tighter than the bootstrapped confidence bounds. Only the theoretically justified confidence bound $L_{n_{opt}}$ of random forest is more conservative. The bias-corrected confidence bounds $L_{bc_{n_{opt}}}$ are always tighter than the bootstrapped confidence bounds. Again, the optimal training size $n_{opt}$ adapts to the saturation level of the curves.

Lasso regression has a smaller PMSE than ridge regression and random forest in this application. Interestingly, the bias-corrected confidence bound ($L_{bc_{n_{opt}}}$) of lasso regression is lower valued than the point estimates of ridge regression and random forest, which suggests a clear difference between the learners.

### 5 Discussion

We presented a novel method, called Learn2Evaluate, to estimate the predictive performance. Learn2Evaluate facilitates the computation of a point estimate and a lower confidence bound, which is proven to control type-I error. This bound may be further tightened by correcting for the estimated bias. This bias-corrected bound is shown to approximately control type-I error in a simulation, although the nominal confidence level is not always guaranteed. Simulations and applications to omics data showed that both bounds are less conservative than a bootstrapped confidence bound. Learn2Evaluate appears to have a lower RMSE of performance point estimates than 10-fold cross-validation and leave-one-out bootstrapping. Furthermore, Learn2Evaluate comes with some additional benefits by providing a dynamic comparison between learners and insight in the
potential benefit of future samples.

One practical limitation of Learn2Evaluate is its computational time because of the relatively large number of splits in a training and test set. This may be a drawback in medium to large sample size settings, because computationally more efficient (asymptotic) methods often suffice [LeDell et al., 2015]. However, for small sample sizes, other resampling techniques also need to generate a large number of splits in a training and test set to yield a reliable performance estimator [Jiang and Simon, 2007, Jiang et al., 2008, Kim, 2009].

In the simulations and the applications, we focused on high dimensional data settings with sample sizes ranging from $N = 75$ to $N = 200$. We focused on omics data by estimating the covariate correlation structure from an omics experiment and by imposing a dense structure on the parameter vector $\beta$. However, Learn2Evaluate applies to any data setting and covariate-response structure. Evaluating the practical usefulness of Learn2Evaluate for such generalizations requires additional simulations.

Results in this study suggest that the power law, which uses all subsample sizes equally, obtains better point estimates than the regression spline, which predominantly uses training set sizes at the end of the learning trajectory to extrapolate. It is therefore interesting to investigate which subsample sizes are most informative for estimating the performance at the full sample size. This may lead to an optimal weighting scheme in the nonlinear least squares fit by an inverse power law, which may improve our point estimates further. Finding an optimal weighting scheme, however, is nontrivial. The variance of the repeated hold-out estimates is difficult to assess because it depends on the complex correlation structure between overlapping training sets [Bengio and Grandvalet, 2004]. We therefore leave this for future research.

Declaration of Interest

The authors declare no competing interests.

Acknowledgement

The authors gratefully acknowledge the financial support by Stichting Hanarth Fonds.
A  Average True AUC

Table A1: Average true AUC for the different simulation settings and learners

|       | Ridge | Lasso | RF |
|-------|-------|-------|----|
| $N = 100, \nu = 1000$ | 0.77  | 0.71  | 0.75 |
| $N = 100, \nu = 100$  | 0.88  | 0.83  | 0.87 |
| $N = 200, \nu = 1000$ | 0.78  | 0.74  | 0.76 |
| $N = 200, \nu = 100$  | 0.89  | 0.86  | 0.87 |

B  Average Training Set Size

Table B1: Average $n_{opt}$ for ridge regression, lasso regression, and random forest (RF).

|       | Ridge | Lasso | RF |
|-------|-------|-------|----|
| $N = 100, \nu = 1000$ | 32    | 45    | 33 |
| $N = 100, \nu = 100$  | 36    | 54    | 34 |
| $N = 200, \nu = 1000$ | 69    | 101   | 76 |
| $N = 200, \nu = 100$  | 76    | 109   | 69 |

C  Supplementary material

Section 1 presents a method to deal with S-shaped learning curves, which may arise as a consequence of a phase transition. Figure S1 shows an instance of a S-shaped learning curve. In Section 2, we show asymptotic variance estimators for the AUC and the PMSE. If such an estimator is not available for the given performance metric, we describe an alternative method in Section 3. Table S1 shows coverage results for this alternative and in Section 4, coverage results for the learning curve fitted by a constrained regression spline are presented. In Section 5, we show boxplots depicting the distance of the confidence bounds of Learn2Evaluate to the true performance (Figures S2, S3, S4, and S5). Section 6 deals with an additional classification application (Figure S6) and in Section 7, Learn2Evaluate is applied in a regression setting. Table S3 (Section 8) shows the stability of the point estimates of Learn2Evaluate with respect to the length.
of the learning trajectory. Finally, Section 9 gives additional details on the data and the software.

References

D. Bamber. The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J. Math. Psychol.*, 12(4):387–415, 1975. doi: 10.1016/0022-2496(75)90001-2.

Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *J. Mach. Learn. Res.*, 5:1089–1105, 2004. URL https://jmlr.csail.mit.edu/papers/v5/grandvalet04a.html.

M. G. Best, N. Sol, I. Kooi, J. Tannous, B. A. Westerman, F. Rustenburg, et al. Rna-seq of tumor-educated platelets enables blood-based pan-cancer, multiclass, and molecular pathway cancer diagnostics. *Cancer cell*, 28(5):666–676, 2015. doi: 10.1016/j.ccell.2015.09.018.

E. A. Boyle, Y. I. Li, and J. K. Pritchard. An expanded view of complex traits: From polygenic to omnigenic. *Cell*, 169(7):1177–1186, 2017. doi: 10.1016/j.cell.2017.05.038.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Mon. Weather. Rev.*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:vofeit⟩2.0.co;2.

P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989. doi: 10.2307/2336116.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

C. Cortes, L. D. Jackel, S. Solla, V. Vapnik, and J. Denker. Learning curves: Asymptotic values and rate of convergence. 6:327–334, 1993. URL https://proceedings.neurips.cc/paper/1993/file/1aa48fc4880bb0c9b8a3bf979d3b917e-Paper.pdf.
E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3):837–845, 1988. doi: 10.2307/2531595.

K. K. Dobbin. A method for constructing a confidence bound for the actual error rate of a prediction rule in high dimensions. *Biostatistics*, 10(2):282–296, 2009. doi: 10.1093/biostatistics/kxn035.

K. K. Dobbin and R. M. Simon. Optimally splitting cases for training and testing high dimensional classifiers. *BMC Med. Genomics.*, 4(1):1–8, 2011. doi: 10.1186/1755-8794-4-31.

D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *P. Natl. Acad. Sci. USA.*, 102(27):9446–9451, 2005. doi: 10.1073/pnas.0502269102.

B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. doi: 10.1201/9780429246593.

N. M. Faber. Estimating the uncertainty in estimates of root mean square error of prediction: application to determining the size of an adequate test set in multivariate calibration. *Chemometr. Intell. Lab.*, 49(1):79–89, 1999. doi: 10.1016/s0169-7439(99)00027-1.

J. Friedman, T. Hastie, and R. J. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1–22, 2010. doi: 10.1163/ej.9789004178922.i-328.7.

J.A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982. doi: 10.1148/radiology.143.1.7063747.

K. R. Hess and C. Wei. Learning curves in classification with microarray data. *Sem. Onc.*, 37(1):65–68, 2010. doi: 10.1053/j.seminoncol.2009.12.002.

H. Ishwaran and U. B. Kogalur. *Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)*, 2021. URL https://cran.r-project.org/package=randomForestSRC.
W. Jiang and R. M. Simon. A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification. *Stat. Med.*, 26(29):5320–5334, 2007. doi: 10.1002/sim.2968.

W. Jiang, S. Varma, and R. M. Simon. Calculating confidence intervals for prediction error in microarray classification using resampling. *Stat. Appl. Genet. Mol.*, 7(1), 2008. doi: 10.2202/1544-6115.1322.

J. H. Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.*, 53(11):3735–3745, 2009. doi: 10.1016/j.csda.2009.04.009.

E. LeDell, M. L. Petersen, and M. J. van der Laan. *Package cvAUC*, 2014. URL https://cran.r-project.org/web/packages/cvAUC/index.html.

E. LeDell, M. L. Petersen, and M. J. van der Laan. Computationally efficient confidence intervals for cross-validated area under the roc curve estimates. *Electron. J. Stat.*, 9(1):1583–1607, 2015. doi: 10.1214/15-ejs1035.

S. Michiels, S. Koscielny, and C. Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *The Lancet*, 365(9458):488–492, 2005. doi: 10.1016/s0140-6736(05)17866-0.

S. Mukherjee, P. Tamayo, S. Rogers, R. Rifkin, A. Engle, C. Campbell, et al. Estimating dataset size requirements for classifying dna microarray data. *J. Comput. Biol.*, 10(2):119–142, 2003. doi: 10.1089/106652703321825928.

P. Ng and M. Maechler. A fast and efficient implementation of qualitatively constrained quantile smoothing splines. *Stat. Model.*, 7(4):315–328, 2007. doi: 10.1177/1471082X0700700403.

P. Ng and M. Maechler. *COBS – Constrained B-splines (Sparse matrix based)*, 2020. URL https://CRAN.R-project.org/package=cobs.

P. W. Novianti, B. C. Snoek, S. M. Wilting, and M. A Van De Wiel. Better diagnostic signatures from rnaseq data through use of auxiliary co-data. *Bioinformatics*, 33(10):1572–1574, 2017. doi: 10.1093/bioinformatics/btw837.
S. Numata, T. Ye, T. Hyde, X. Guitart, R. Tao, M. Wininger, et al. Dna methylation signatures in development and aging of the human prefrontal cortex. *Am. J. Hum. Genet.*, 90:260–72, 2012. doi: 10.1016/j.ajhg.2011.12.020.

X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J. C. Sanchez, and M. Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12(1):77, 2011. doi: 10.1186/1471-2105-12-77.

J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol.*, 4(1), 2005. doi: 10.2202/1544-6115.1175.

J. Schäfer, R. Opgen-Rhein, V. Zuber, M. Ahdesmaki, A. P. D. Silva, and K. Strimmer. *Package corpcor*, 2017. URL https://CRAN.R-project.org/package=corpcor.

M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Roy. Stat. Soc. B. Met.*, 36(2):111–133, 1974. doi: 10.1111/j.2517-6161.1974.tb00994.x.

M. A. van de Wiel, J. Berkhof, and W. N. van Wieringen. Testing the prediction error difference between 2 predictors. *Biostatistics*, 10(3):550–560, 2009. doi: 10.1093/biostatistics/kxp011.
1 S-shaped learning curves

In some situations, the concavity or convexity constraint on the learning curve may be inadequate. An example of such a situation is the phase transition of lasso regression when the true covariate-response structure is sparse and the outcome continuous [Donoho and Tanner, 2005, 2009]. Here, the performance, quantified by the mean square error of prediction (PMSE), follows a reverse S-shaped curve as a function of the training set size. In the first phase, the curve is decreasing and concave, and in the second phase, starting after the inflection point, the curve is decreasing and convex.

For completeness, we extend Learn2Evaluate to also deal with S-shaped curves. Users should visually inspect whether such a curve fit is appropriate by plotting the learning trajectory (eq. 2 in main document). For phase transitions, we are mainly interested in the second phase because the curves are used to extrapolate to the full sample size $N$.

Therefore, we start by estimating the inflection point $n_{tr}$, after which the curve will have its familiar behavior: increasing and concave for increasing performance metrics such as the AUC, and decreasing and convex for decreasing performance metrics such as the mean square error of prediction (PMSE). To determine $n_{tr}$, we fit a generalized logistic curve [Richards, 1959], which is defined by

$$f(n) = A + \frac{K - A}{(1 + \exp(-B(n - M)))^{1/\nu}}. \quad (1)$$

*Corresponding author, E-mail address: j.m.goedhart@amsterdamumc.nl
with $A$ being the left asymptote, $K$ the right asymptote, and $B$ the growth rate. Parameters $M$ and $\nu$ are related to the (a)symmetry of the curve. Note that (1) holds for both decreasing and increasing performance metrics by modifying the parameters. To estimate the parameters, we solve

$$
(\hat{A}, \hat{B}, \hat{K}, \hat{M}, \hat{\nu}) = \arg \min_{A, B, K, M, \nu} \sum_{j=1}^{J} \left( A + \frac{K - A}{(1 + \exp (-B(n_j - M)))^{1/\nu}} - \hat{\rho}_{n_j} \right)^2,
$$

using the base R optim function. The estimated inflection point $\hat{n}_{tr}$ is then given by

$$
\hat{n}_{tr} = \hat{M} - \frac{1}{\hat{B}} \log \hat{\nu},
$$

as derived in [Taye and Purnachandra, 2013]. After $\hat{n}_{tr}$ is determined, we fit the standard learning curve using $\hat{n}_{tr}$ as minimal subsample size, i.e. $n_1 = \hat{n}_{tr}$.

### 1.1 Illustration on simulated data

Here, we illustrate the above described methodology on simulated data for lasso regression and the PMSE as performance metric.

The $i$th instance of response variable $Y$ is defined by

$$
Y_i = X_i \beta + \epsilon, \quad i = 1, \ldots, N,
$$

with $\epsilon \sim N(0, 0.2)$ and $N = 100$. The $p = 1000$ covariates $X_i$ are independently drawn from a standard normal distribution, $X_i \sim N(0, 1)$. For the sparse parameter vector $\beta$, we set the first five entries to 0.6 and the remaining entries are fixed at zero.

We construct learning curves (Figure 1) for lasso regression by first estimating a learning trajectory of 20 different subsample sizes $n_j$ in the regime $[10, 90]$ by applying repeated (50 repeats) hold-out estimation (blue dots). The penalty parameter of lasso regression is estimated once per $n_j$, which is then fixed for the 50 repeats of each $n_j$. We estimate this penalty parameter by the median of repeated (5 times) 10-fold CV. We fit the learning trajectory using (2) (blue solid line) and we estimate the inflection point $n_{tr}$ using (3) (black dotted vertical line). After $\hat{n}_{tr}$ is determined, we estimate the learning trajectory in the subsample regime $[\hat{n}_{tr}, 90]$ (black dots) using the same procedure as above. We fit this learning trajectory using the inverse power law for decreasing and convex data (black solid line).
Figure 1 illustrates that the generalized logistic function, i.e. (1), describes phase transitions for lasso regression well. Furthermore, the inverse power law and (1) show good agreement after the estimated inflection point $\hat{n}_{tr}$.

Although in this case the differences between (1) and the decreasing convex inverse power law are negligible, we prefer to fit a power law after the estimated inflection point. The second phase of the learning trajectory is of interest to estimate the performance and a training set size for the confidence bound. In this way, the potential influence of the first phase of the learning trajectory on these estimates is avoided.

2 Asymptotic variance estimators

To minimize the mean square error of the predictive performance estimate with respect to the subsample size $n$ (eq. 11 of the main document), an asymptotic variance estimator is required. Here, we show asymptotic variance estimators for the AUC and the PMSE.

For the AUC, Bamber [1975] derived an estimator, which is given by

$$\hat{\text{Var}}_n(AUC) = \frac{Q_1 + Q_2 + Q_3}{n_+ n_-},$$

(5)
with
\[ Q_1 = \hat{A}UC_n (1 - \hat{A}UC_n), \quad Q_2 = (n_+ - 1) \left( \frac{\hat{\rho}_n}{2 - \hat{A}UC_n} - \hat{A}UC_n^2 \right), \quad \text{and} \]
\[ Q_3 = (n_- - 1) \left( \frac{2\hat{A}UC_n^2}{1 + \hat{A}UC_n} - \hat{A}UC_n^2 \right). \]

Here, \( n_+ \) denotes the number of positive cases (\( Y = 1 \)) and \( n_- = n - n_+ \) the number of negative cases (\( Y = 0 \)) in the test set of size \( n \).

For the PMSE, we use the variance estimator derived by Faber [1999], which reads
\[ \hat{\text{Var}}(\text{PMSE}) = \frac{2\text{PMSE}^2_n}{n}. \] (6)

### 3 Alternative method to MSE minimization

#### 3.1 Method

To determine an optimal training set size \( n_{opt} \) when an asymptotic performance variance estimator such as (5) or (6) is not known, we suggest to control for the (empirical) bias. Specifically, we define an acceptable margin \( m \) and we find the smallest subsample size \( n \) for which we have
\[ n_{opt} = \arg \min_n \{ n : f(N) - f(n) \leq m \}. \] (7)

This method, which we call bias margin control, ensures that we obtain a maximum amount of test samples, while controlling the empirical bias. A drawback of this method is that the choice of \( m \) is arbitrary.

#### 3.2 Simulation results

Table 1 shows coverage results for the 95% confidence bounds of Learn2Evaluate with \( (L_{n_{opt}}^{bc}) \) and without \( (L_{n_{opt}}) \) bias correction. Here, we determine \( n_{opt} \) by bias margin control, i.e. (7), using a margin of 2% of the point estimate \( f(N) \). For completeness, we also show coverage results for the asymptotic estimator derived by LeDell et al. [2015] (Le Dell) and leave-one-out bootstrapping (LOOB).
Table 1: Coverage results for the 95% lower confidence bound of Learn2Evaluate without bias correction ($L_{n_{opt}}$), Learn2Evaluate with bias correction ($L_{bc,n_{opt}}^{bc}$), the asymptotic AUC variance estimator of Le Dell (Le Dell), and leave-one-out bootstrapping (LOOB). We use bias margin control to determine $n_{opt}$. Learning curves are fitted by inverse power laws.

| Method   | Ridge | Lasso | RF | Ridge | Lasso | RF |
|----------|-------|-------|----|-------|-------|----|
| $N = 100$, $L_{n_{opt}}$ | 0.974 | 0.988 | 0.979 | $L_{n_{opt}}$ | 0.976 | 0.985 | 0.983 |
| $L_{bc,n_{opt}}^{bc}$ | 0.954 | 0.965 | 0.935 | $N = 100$, $L_{bc,n_{opt}}^{bc}$ | 0.937 | 0.977 | 0.945 |
| $\nu = 1000$ Le Dell | 0.895 | 0.851 | 0.876 | $\nu = 1000$ Le Dell | 0.870 | 0.847 | 0.892 |
| LOOB | 0.990 | 0.996 | 0.990 | LOOB | 0.993 | 0.995 | 0.995 |
| $N = 200$, $L_{n_{opt}}$ | 0.992 | 0.993 | 0.992 | $L_{n_{opt}}$ | 0.994 | 0.995 | 0.979 |
| $L_{bc,n_{opt}}^{bc}$ | 0.972 | 0.985 | 0.971 | $N = 200$, $L_{bc,n_{opt}}^{bc}$ | 0.952 | 0.973 | 0.928 |
| $\nu = 1000$ Le Dell | 0.911 | 0.874 | 0.884 | $\nu = 1000$ Le Dell | 0.908 | 0.886 | 0.869 |
| LOOB | 0.996 | 1.00 | 0.992 | LOOB | 0.996 | 0.998 | 0.989 |

Table 1 shows similar results as in Table 2 of the main document. The theoretically justified lower confidence bound $L_{n_{opt}}$ controls the type-I error well below the nominal level and is less conservative than leave-one-out bootstrapping. Coverage is closer to nominal level when a bias correction is performed ($L_{bc,n_{opt}}^{bc}$). Again, in setting $N = 200$, $\nu = 100$, $L_{bc,n_{opt}}^{bc}$ of random forest is too liberal when taking the binomial error of the coverage estimate into account.

4 Coverage results for the spline

Table 2 shows the coverage of the 95% lower confidence bounds $L_{n_{opt}}$ and $L_{n_{opt}}^{bc}$ of Learn2Evaluate, with learning curves fitted by a constrained regression spline. We show results for both methods to determine an optimal training set size $n_{opt}$, i.e. bias margin control (Bias, Equation 7) and MSE minimization (MSE, eq. 11 of the main document).
Table 2: Coverage results for the 95% lower confidence bound of Learn2Evaluate without bias correction ($L_{n_{opt}}$) and Learn2Evaluate with bias correction ($L_{n_{opt}}^{bc}$). We use bias margin control (Bias) and MSE minimization (MSE) to determine $n_{opt}$. Learning curves are fitted by constrained regression splines.

|         | Ridge | Lasso | RF |         | Ridge | Lasso | RF |
|---------|-------|-------|----|---------|-------|-------|----|
|         | $L_{n_{opt}}$ Bias | 0.977 | 0.987 | 0.983 | $L_{n_{opt}}$ Bias | 0.978 | 0.983 | 0.982 |
| $N = 100$, | $L_{n_{opt}}$ MSE | 0.976 | 0.979 | 0.983 | $N = 100$, | $L_{n_{opt}}$ MSE | 0.978 | 0.959 | 0.985 |
| $\nu = 1000$ | $L_{n_{opt}}^{bc}$ Bias | 0.938 | 0.925 | 0.923 | $\nu = 1000$ | $L_{n_{opt}}^{bc}$ Bias | 0.926 | 0.961 | 0.942 |
|         | $L_{n_{opt}}^{bc}$ MSE | 0.961 | 0.952 | 0.945 | | $L_{n_{opt}}^{bc}$ MSE | 0.938 | 0.950 | 0.942 |
| $N = 200$, | $L_{n_{opt}}$ Bias | 0.991 | 0.993 | 0.990 | $N = 200$, | $L_{n_{opt}}$ MSE | 0.991 | 0.991 | 0.983 |
| $\nu = 1000$ | $L_{n_{opt}}^{bc}$ Bias | 0.959 | 0.961 | 0.970 | $\nu = 1000$ | $L_{n_{opt}}^{bc}$ Bias | 0.957 | 0.905 | 0.923 |
|         | $L_{n_{opt}}^{bc}$ MSE | 0.976 | 0.987 | 0.975 | | $L_{n_{opt}}^{bc}$ MSE | 0.952 | 0.969 | 0.925 |

5 Boxplots of confidence bound distances

Here, we depict, for all simulation settings, boxplots of the distance of the lower confidence bound to the corresponding target parameter AUC$_r$ for the bootstrap method (gray), and Learn2Evaluate with (L2E + BC, brown), and without (L2E - BC, green) bias correction. The learning curve is fitted by an inverse power law (eq. 4 of the main document) and $n_{opt}$ is determined by MSE minimization (eq. 11 of the main document).
Figure 2: Boxplots of the empirical distribution of distances of the lower 95% confidence bounds to the true $AUC$ values (y-axis) for the bootstrap, and Learn2Evaluate without (L2E - BC) and with bias correction (L2E + BC) in the $N = 100, \nu = 1000$ setting. The learning curve is fitted by an inverse power law and we determine $n_{opt}$ by MSE minimization. Outliers are not depicted.

Figure 3: Boxplots of the empirical distribution of distances of the lower 95% confidence bounds to the true $AUC$ values (y-axis) for the bootstrap, and Learn2Evaluate without (L2E - BC) and with bias correction (L2E + BC) in the $N = 100, \nu = 100$ setting. The learning curve is fitted by an inverse power law and we determine $n_{opt}$ by MSE minimization. Outliers are not depicted.
Figure 4: Boxplots of the empirical distribution of distances of the lower 95% confidence bounds to the true $AUC$ values (y-axis) for the bootstrap, and Learn2Evaluate without ($L2E - BC$) and with bias correction ($L2E + BC$) in the $N = 200, \nu = 1000$ setting. The learning curve is fitted by an inverse power law and we determine $n_{opt}$ by MSE minimization. Outliers are not depicted.

Figure 5: Boxplots of the empirical distribution of distances of the lower 95% confidence bounds to the true $AUC$ values (y-axis) for the bootstrap, and Learn2Evaluate without ($L2E - BC$) and with bias correction ($L2E + BC$) in the $N = 200, \nu = 1000$ setting. The learning curve is fitted by an inverse power law and we determine $n_{opt}$ by MSE minimization. Outliers are not depicted.
6 Three other applications of Learn2Evaluate

Here, we depict some additional learning curves in a classification setting. Learn2Evaluate is again applied to mRNA-Seq data extracted from blood platelets. The same methodology as for Figure 2 in the main document is used. We consider the following binary classifications: non-small-cell lung cancer (NSCLC, \( n = 60 \)) versus pancreas cancer (Pancreas, \( n = 35 \)) using \( p = 18,582 \) transcripts (Figure 6a), breast cancer (Breast, \( n = 40 \)) versus colorectal cancer (CRC, \( n = 41 \)) using \( p = 18,410 \) transcripts (Figure 6b), and colorectal cancer (\( n = 41 \)) versus pancreas cancer (\( n = 35 \)) using \( p = 18,090 \) transcripts (Figure 6c).
Figure 6: Learning curves for three classification tasks. In each task, we produce a learning curve for a ridge (black), a lasso (blue), and a random forest (green) model. For each learner, we depict the learning trajectory (dots) together with the inverse power law fit (solid line). We show confidence bounds of Learn2Evaluate with ($L_{n_{opt}}^\text{bc}$, downward triangle) and without ($L_{n_{opt}}$, upward triangle) bias correction, and bootstrapping (Bootstrap, square). We use MSE minimization to determine $n_{opt}$ (dashed vertical line).

7 Regression application

To apply Learn2Evaluate in a regression setting, we use a DNA methylation data set described by Akalin [2020]. Experiments have demonstrated that the methylation level of CpG islands are predictive of age [Numata et al., 2012, Horvath, 2013]. The processed data set is available via https://github.com/JeroenGoedhart/Learn2Evaluate. Data preprocessing is described in Section 9
We generate learning curves for ridge regression (black), lasso regression (blue), and random forest (green) (Figure 7). Learning curves are obtained by the same procedure as in the simulation study and the application to mRNA-Seq data. For ten homogeneously spaced subsample sizes, we estimate the predictive performance, quantified by the PMSE, by repeated (50 times) hold-out estimation. This renders the learning trajectory (dots, eq. 2 of the main document). The maximum allowed subsample size equals $N - 10$. We then fit an inverse power law (solid line, eq. 4 of the main document), which is decreasing and convex.

We construct upper confidence bounds based on Learn2Evaluate without bias correction ($L_{n_{opt}}$, upward triangle), Learn2Evaluate with bias correction ($L_{n_{opt}}^{bc}$, downward triangle), and leave-one-out bootstrapping (Bootstrap, square). We use MSE minimization (eq. 11 of the main document) to determine an optimal training subsample size $n_{opt}$ (dashed vertical line). We use (6) to estimate the variance of the PMSE and to construct an 95% upper confidence bound. We also display the point estimates of Learn2Evaluate (star).

Figure 7 illustrates that the decreasing and convex power law fits the learning trajec-

Figure 7: Learning curves for predicting age from DNA methylation data. We produce a learning curve for a ridge (black/dark), a lasso (blue/medium), and a random forest (green/light) model. For the learning curve, we depict the learning trajectory (dots) together with the inverse power law fit (solid lines). For each learner, we show confidence bounds constructed by Learn2Evaluate without bias correction ($L_{n_{opt}}$, upward triangle), Learn2Evaluate with bias correction ($L_{n_{opt}}^{bc}$, downward triangle), and bootstrapping (Bootstrap, square). We used MSE minimization to determine $n_{opt}$ (dashed vertical line).

Figure 7 illustrates that the decreasing and convex power law fits the learning trajec-
tories for the PMSE well. The upper confidence bounds of Learn2Evaluate are in most cases tighter than the bootstrapped confidence bounds. Only the theoretically justified confidence bound $L_{n_{opt}}$ of random forest is more conservative. The bias-corrected confidence bounds $L_{n_{opt}}^{bc}$ are always tighter than the bootstrapped confidence bounds. Again, the optimal training sizes $n_{opt}$ nicely adapt to the saturation level of the curves.

Lasso regression has a smaller PMSE than ridge regression and random forest in this application. Interestingly, the bias-corrected confidence bound $(L_{n_{opt}}^{bc})$ of lasso regression is lower-valued than the point estimates of ridge regression and random forest, which suggests a clear difference between the learners.

8 Varying the learning trajectory

Here, we show the stability of the AUC point estimates of Learn2Evaluate, obtained in the application to blood-platelet mRNA-Seq data. In Table 3, we depict these point estimates when a learning trajectory (eq. 2 in the main document) of 10 subsample sizes is considered and when a learning trajectory of 20 subsample sizes is considered. In both scenarios, the number of repeats for each subsample size is fixed at 50.
Table 3: Stability of AUC point estimates for a varying length of the learning trajectory (10 and 20 subsample sizes).

|                          | Ridge | Lasso | Random Forest |
|--------------------------|-------|-------|---------------|
| **Breast vs. NSCLC**     |       |       |               |
| 10 subsample sizes       | 0.826 | 0.840 | 0.829         |
| 20 subsample sizes       | 0.823 | 0.835 | 0.834         |
| **Breast vs. CRC**       |       |       |               |
| 10 subsample sizes       | 0.843 | 0.828 | 0.847         |
| 20 subsample sizes       | 0.849 | 0.833 | 0.850         |
| **Breast vs. Pancreas**  |       |       |               |
| 10 subsample sizes       | 0.888 | 0.918 | 0.856         |
| 20 subsample sizes       | 0.890 | 0.922 | 0.853         |
| **NSCLC vs. Control**    |       |       |               |
| 10 subsample sizes       | 0.975 | 0.962 | 0.967         |
| 20 subsample sizes       | 0.974 | 0.963 | 0.967         |
| **NSCLC vs. Pancreas**   |       |       |               |
| 10 subsample sizes       | 0.954 | 0.948 | 0.944         |
| 20 subsample sizes       | 0.954 | 0.949 | 0.943         |
| **CRC vs. Pancreas**     |       |       |               |
| 10 subsample sizes       | 0.749 | 0.618 | 0.726         |
| 20 subsample sizes       | 0.750 | 0.622 | 0.723         |

Table 3 shows that increasing the size of the learning trajectory to 20 subsample sizes does hardly change the performance estimates. The largest difference (0.6%) is observed for lasso regression in the CRC vs. Pancreas classification task.

9 Details on data and software

9.1 Details on mRNA-Seq data set

RNA profiles, extracted from bloodplatelets, were obtained from 285 individuals, as described in Best et al. [2015]. Of those individuals, 55 belong to the control group, and the
remaining individuals have one of the in total six tumor types: breast cancer \((n = 40)\), colocteral cancer \((n = 41)\), glioblastoma \((n = 40)\), hepatobiliary cancer \((n = 14)\), non-small-cell lung carcinoma \((n = 60)\), and pancreatic cancer \((n = 35)\). Raw sequencing data are available in the GEO database (GEO: GSE68086) and in the ArrayExpress repository (ArrayExpress ID: E-GEOD-68086). The raw data set has measurements of 57,736 transcripts.

Data preprocessing is described by Novianti et al. [2017] in the supplementary material. We summarize their work here. Raw count data were first normalized by weighted trimmed mean of M-values [Robinson and Oshlack, 2010]. Next, transcripts having a count less than 5 across samples were filtered out, rendering 19,629 transcripts. Finally, the data were transformed to a quasi-gaussian scale, i.e. \(\sqrt{x_{ij} + \frac{3}{8}} - \sqrt{\frac{3}{8}}\), where \(x_{ij}\) denotes the count of the \(j\)th transcript in the \(i\)th sample. The processed data set is available via https://github.com/JeroenGoedhart/Learn2Evaluate. Prior to building the classifiers, we excluded transcripts with zero variance and we standardized the data.

9.2 Details on DNA methylation data set

The DNA methylation data set is described in [Akalin, 2020] and can be found in the R package compGenomRData. Methylation levels of \(p = 27579\) CpG islands were measured for 108 individuals. Covariates having a standard deviation below 0.1 were removed, which renders a total of \(p = 2290\) covariates. Next, the data was standardized, rendering the processed data set, which is available via https://github.com/JeroenGoedhart/Learn2Evaluate.

9.3 Details on software

R code to reproduce results is available via https://github.com/JeroenGoedhart/Learn2Evaluate. We use R version 4.0.3. The repository Learn2Evaluate contains the data map, where the processed RNA sequencing and DNA methylation data are located, the simulation-results map, which contains R scripts to reproduce the simulation results in the paper (including seeds for the pseudo-random-number generator), and the R scripts map. The latter map includes the R script “Learn2Evaluate”, which contains the function to apply Learn2Evaluate to a data set of choice. Details are in the R script. The R
scripts map also contains the script “run_Learn2_Evaluate_real_data”, which displays the
code to reproduce the results in the application section.

The software depends on the following R packages: mvtnorm (version 1.1-3), glmnet
(version 4.1-3), cobs (version 1.3-4), randomForestSRC (version 3.0.2), pROC (version
1.18.1), and cvAUC (version 1.1.0).

References

Altuna Akalin. *Computational Genomics with r*. Chapman and Hall/CRC, 2020. doi:
https://doi.org/10.1201/9780429084317.

D. Bamber. The area above the ordinal dominance graph and the area below the receiver
operating characteristic graph. *J. Math. Psychol.*, 12(4):387–415, 1975. doi: 10.1016/
0022-2496(75)90001-2.

M. G. Best, N. Sol, I. Kooi, J. Tannous, B. A. Westerman, F. Rustenburg, et al. Rna-seq
of tumor-educated platelets enables blood-based pan-cancer, multiclass, and molecular
pathway cancer diagnostics. *Cancer cell*, 28(5):666–676, 2015. doi: 10.1016/j.ccell.
2015.09.018.

D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear
equations by linear programming. *P. Natl. Acad. Sci. USA.*, 102(27):9446–9451, 2005.
doi: 10.1073/pnas.0502269102.

D. L. Donoho and J. Tanner. Observed universality of phase transitions in high-
dimensional geometry, with implications for modern data analysis and signal process-
ing. *Philos. T. Roy. Soc. A.*, 367(1906):4273–4293, 2009. doi: 10.1098/rsta.2009.0152.

N. M. Faber. Estimating the uncertainty in estimates of root mean square error of
prediction: application to determining the size of an adequate test set in multivariate
calibration. *Chemometr. Intell. Lab.*, 49(1):79–89, 1999. doi: 10.1016/s0169-7439(99)
00027-1.

S. Horvath. Dna methylation age of human tissues and cell types. *Genome Biol.*, 14(10):
1–20, 2013. doi: https://doi.org/10.1186/gb-2013-14-10-r115.
E. LeDell, M. L. Petersen, and M. J. van der Laan. Computationally efficient confidence intervals for cross-validated area under the roc curve estimates. *Electron. J. Stat.*, 9 1 (1):1583–1607, 2015. doi: 10.1214/15-ejs1035.

P. W. Novianti, B. C. Snoek, S. M. Wilting, and M. A Van De Wiel. Better diagnostic signatures from rnaseq data through use of auxiliary co-data. *Bioinformatics*, 33(10): 1572–1574, 2017. doi: 10.1093/bioinformatics/btw837.

S. Numata, T. Ye, T. Hyde, X. Guitart, R. Tao, M. Wininger, et al. Dna methylation signatures in development and aging of the human prefrontal cortex. *Am. J. Hum. Genet.*, 90:260–72, 2012. doi: 10.1016/j.ajhg.2011.12.020.

F. J. Richards. A flexible growth function for empirical use. *J. Exp. Bot.*, 10(2):290–301, 06 1959. doi: 10.1093/jxb/10.2.290.

M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biol.*, 11:R25, 03 2010. doi: 10.1186/gb-2010-11-3-r25.

A. Taye and R. K. Purnachandra. Derivation of inflection points of nonlinear regression curves - implications to statistics. *Am. J. Theor. Appl. Stat.*, 2:268, 01 2013. doi: 10.11648/j.ajtas.20130206.25.