BMJ Open

One-step extrapolation of the prediction performance of a gene signature derived from a small study

Ling-Yi Wang,[1,2] Wen-Chung Lee[1]

To cite: Wang L-Y, Lee W-C. One-step extrapolation of the prediction performance of a gene signature derived from a small study. *BMJ Open* 2015;**5**:e007170. doi:10.1136/bmjopen-2014-007170

▸ Prepublication history and additional material is available. To view please visit the journal (http://dx.doi.org/10.1136/bmjopen-2014-007170).

Received 14 November 2014
Revised 4 March 2015
Accepted 20 March 2015

[1]Research Center for Genes, Environment and Human Health, and Institute of Epidemiology and Preventive Medicine, College of Public Health, National Taiwan University, Taipei, Taiwan
[2]Department of Medical Research, Tzu Chi General Hospital, Hualien, Taiwan

Correspondence to
Professor Wen-Chung Lee; wenchung@ntu.edu.tw

ABSTRACT

Objective: Microarray-related studies often involve a very large number of genes and small sample size. Cross-validating or bootstrapping is therefore imperative to obtain a fair assessment of the prediction/classification performance of a gene signature. A deficiency of these methods is the reduced training sample size because of the partition process in cross-validation and sampling with replacement in bootstrapping. To address this problem, we aim to obtain a prediction performance estimate that strikes a good balance between bias and variance and has a small root mean squared error.

Methods: We propose to make a one-step extrapolation from the fitted learning curve to estimate the prediction/classification performance of the model trained by all the samples.

Results: Simulation studies show that the method strikes a good balance between bias and variance and has a small root mean squared error. Three microarray data sets are used for demonstration.

Conclusions: Our method is advocated to estimate the prediction performance of a gene signature derived from a small study.

Strengths and limitations of this study

- The proposed method estimates the prediction performance of a gene signature derived from a small study.
- The proposed method strikes a good balance between bias and variance and has a small root mean squared error.
- The proposed method can be applied to linear and non-linear prediction models.
- The proposed method may not work well for studies with an extremely small sample size (eg, n<10).

INTRODUCTION

With the advances in microarray technology, hundreds of thousands of genes with expression information on an individual can be obtained in a single experiment. This high throughput technology enables us to make diagnostic and prognostic predictions based on a participant's gene signature.[1–14] There are four key steps in microarray-based studies: (1) data processing (eg, data normalisation), (2) gene selection, (3) prediction model construction and (4) prediction performance evaluation.[15] This paper focuses on the last step, the evaluation of prediction performances.

Microarray-based studies often involve a very large number of genes and a relatively small sample size. The same small data set being used for constructing the prediction model and subsequently evaluating the prediction performance tends to give over-optimistic estimates. This is why cross-validating or bootstrapping is imperative if a fair assessment of the prediction/classification performance of a gene signature is to be made. Popular cross-validation (CV) methods are k-fold CV, Monte Carlo CV and leave-one-out CV (LOOCV, also known as jackknifing).[16] These methods partition the original data into a training set and a testing set. The training sample size, therefore, is reduced. The bootstrap method is an alternative to CV that operates by sampling with replacement of the original data.[17 18] Even though the bootstrap sample has the same sample size as the original one, the overlapping of subjects between the bootstrap sample and the original data still reduces the effective (non-overlapping) training sample size. The reduced training sample size will curtail the prediction/classification performance of a gene signature, especially when the sample size of a study is already small.[19]

Estimating the performance of a prediction model built from a small study is a vexing task. For CV purposes, the already small sample size still needs to be partitioned further into a training set and a testing one. If we make the training size as large as possible (such as LOOCV), it would be nearly unbiased, but the effective sample size left for validation is one subject and we would get a variance that is unduly large, that is, the notorious bias–variance dilemma.[20 21]

The accuracy rate, the error rate and the area under the receiver operating characteristic curve (AUC) are commonly used performance indicators of a prediction model for a binary outcome.[15 22] In this paper, we focus on the AUC index because it evaluates the global performance of a prediction model, not just at a particular cut-off point, but for each and every possible cut-off point. In a small data set, each and every subject is indispensable. We propose to make a one-step extrapolation from the fitted learning curve for AUC to estimate the prediction/classification performance of the model trained by all the samples.

METHODS

Monte Carlo CVs

Suppose that a given prediction model is to be evaluated based on a data set with a total of N_1 cases (or individuals with adverse events) and N_0 controls (or individuals without adverse events). First, we use CVs with five different folds (LOOCV, 10-fold, 5-fold, 3-fold and 2-fold CV, respectively) to evaluate the performances of the prediction model. (LOOCV is the largest training size possible and the 2-fold CV is the smallest. Between these two extremes, we add three additional fold numbers. More folds can also be tried, but the results are similar.) To be more precise, we use the Monte Carlo random partition to partition the data set into two parts: the training set and the testing set. The random partitions are operated separately for the case group and the control group to ensure that the number of cases and controls is as balanced as possible;[23 24] the training set has a total of $n_1 = N_1 - (N_1/k)$ cases and $n_0 = N_0 - (N_0/k)$ controls (both to the nearest integers), where $k = N_1(N_0)$, 10, 5, 3, 2 for LOOCV, 10-fold, 5-fold, 3-fold and 2-fold CVs, respectively.

To be representative of all possible partitions, we suggest running at least 100 Monte Carlo random partitions for all folds, although this may be superfluous for some situations. (For example, there are only $8 \times 8 = 64$ distinctive partitions for LOOCV with $N_1 = N_0 = 8$. Note that in a random partition, we leave out 'one pair' of a case and a control, instead of 'one subject'.[24]) For each Monte Carlo partition, a prediction model will be built from the training set, and the testing set used to evaluate the performance of this model. The AUCs under the same fold are to be averaged (denoted as $\overline{\text{AUC}}$), which leaves us with a total of five $\overline{\text{AUC}}$s.

Learning curve for AUC

A learning curve is an assumed functional relation between prediction performance and training sample size.[25] In this study, we take $y = a + bx$ as our learning curve, where $y = Z_{\text{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$. Essentially, this learning curve is a straight line in a double-inverse coordinate. For the ordinate, the AUC values are first transformed to quantiles of standard normal distribution, then squared and finally inversed. This expands the range of 0.5–1.0 in an AUC to a range of $0 - \infty$ in the ordinate. For the abscissa, the range of the sample size is also between 0 and ∞. Such a sample size in inverse should be more sensitive to changes when the original sample size is small. The online supplementary appendix 1 shows that this learning curve is the exact functional relation between prediction performance and training sample size when normality and independence of the data are assumed.

One-step extrapolation from the learning curve

We calculate $y = Z_{\text{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$ for each fold. On the basis of the five coordinate points, (x, y)s, we draw a linear regression line, that is, $y = a + bx$. To extrapolate the performance (denoted as AUC_T) of the prediction model when all the subjects $N_T = N_1 + N_0 (N_T = N_1 + N_0)$ are used as training samples, we enter N_1, N_0 into the equation to get $Z_{\text{AUC}_T}^{-2} = a + b \times (N_1^{-1} + N_0^{-1})$.

With $\hat{y} = Z_{\text{AUC}_T}^{-2}$ calculated, we then obtain $\widehat{\text{AUC}}_T = \Phi\left(\sqrt{\hat{y}^{-1}}\right)$, where $\Phi(\cdot)$ is the cumulative distribution for the standard normal.

SIMULATION STUDIES

Data and prediction models

We consider four different sample sizes: $N_T = 10$ (cases) +10 (controls), 15+15, 20+20 and 25+25, respectively. A total of 10 genes are considered. The gene expression levels are generated from a normal distribution with a variance of 1. For the cases, the means of the gene expressions are distributed as uniform (−0.8, 0.8); for the controls, the means are set to 0.

Four different data structures are considered. The first three are normally distributed with a correlation coefficient of 0 (independence), 0.2 and 0.5 (dependency). The last data structure is more complicated. For the cases, the expression level of each gene is distributed as a mixture of three normal distributions with variances of 1. The three means are generated from a uniform (−1.5, 1.5) distribution with a probability of 0.6, a uniform (−1.2, 1.2) distribution with a probability of 0.3 and a uniform (−1, 1) distribution with a probability of 0.1, respectively (see online supplementary appendix 2 for this non-normal distribution). The gene expression level for the controls follows the standard normal distribution. The correlation coefficient between any two genes is set at 0.5 in these complex data.

There are many methods to build prediction models. In this paper, we use the naïve multiple regression and the support vector machine (SVM), as detailed below, to build prediction models. Another machine learning method, the random forest (RF), is detailed in online supplementary materials.

The naïve multiple regression is a simple prediction method. First, the β-coefficients ($\hat{\beta}_i, i = 1, \ldots, p$ where p is the number of genes in the gene signature) are calculated as the mean expression difference for each gene

between the case and the control groups in the training set. The prediction score of the naïve multiple regression for the jth subject in the testing set is then $\Sigma_{i=1}^{p}\widehat{\beta}_i\chi_{ij}$, where χ_{ij} is the observed gene expression level of the ith gene for this jth subject. (The naïve multiple regression used in this study is similar to a previously proposed compound covariate method[26] where the prediction score for the jth subject in the testing set is $\Sigma_{i=1}^{p}t_i\chi_{ij}$ with the two-sample t-statistic of each gene serving its own weight in the prediction model).

SVM is a more sophisticated method; it is a very efficient learning algorithm for high-dimensional data in classification, regression and pattern recognition. The basis of SVM is to implicitly map data to a higher dimensional space via a kernel function in order to identify an optimal hyperplane that maximises the margin between the two groups.[27] There are many software packages available to implement SVM. In this study, we use the e1071-package of R with a default radial basis function kernel to obtain the prediction scores.[28]

CVs of the prediction models

In our simulation study, we perform a total of 5000 simulations. In each simulation, a total of 100 random partitions are performed for each fold CV (LOOCV, 10-fold, 5-fold, 3-fold and 2-fold CVs, respectively). From these, we use the previously described learning curve to make a one-step extrapolation to the cross-validated AUC when all the samples are utilised to train the model. For a comparison, we also calculate the internally validated AUCs of the LOO bootstrap in each simulation. This is a modified bootstrap procedure of the ordinary bootstrap. We draw a total of 100 resamplings. At each draw, the observations left out serve as the testing set. The effective (non-overlapping) training sample size of the LOO bootstrap is around 63.2% of the total sample size.[17 18] (Out-of-bag (OOB)[29] estimation employs a majority vote on the multiple prediction made for observation i based on the bootstrap samples at each draw, while the LOO bootstrap takes an average on error of these predictions. Therefore, OOB estimation may have larger variability than the LOO bootstrap when the sample size is small.[30]) In the simulation, we additionally create a large data set of 1000 cases and 1000 controls for external validation. For a prediction model, an externally validated AUC against this data set is considered as its true AUC value (one true AUC for each round of simulation). It should be pointed out that in real practice, one rarely has the luxury to conduct such a large-scale external validation, but will often have to settle for a satisfactory internal validation method which is precisely the focal point of this paper.

Bias, variance and root mean squared error

In each round of the simulation, we calculate an estimated AUC, an error (the difference between the estimated AUC and the true AUC) and an error square for each performance evaluation method. On the basis of the 5000 simulations, the bias is calculated as the sample mean of the errors; the variance is the sample variance of the estimated AUCs; and the mean squared error (MSE) is the sample mean of the error squares. Finally, the root mean squared error (RMSE) is calculated from the square root of MSE. (RMSE simultaneously considers bias and variance. This value represents the 'average' (root-mean-square average, to be precise) difference between the estimated AUC and the true AUC).

Simulation results

In figure 1, we present the bias (panels A–D), the variance (panels E–H) and the RMSE (panels I–L), respectively, using naïve multiple regression under different sample sizes. When the variables are independently distributed (panels A, E and I), the bias becomes smaller as the sample size becomes larger (closer to zero; panel A). All the fold-based CV methods underestimate the true AUC value because the training sample sizes they use are smaller than the total sample size given. The training sample size of LOOCV is closest to the total sample size (total sample size minus one pair); hence, it is the least biased (blue line) among all the fold-based CV methods. The training sample size of the LOO bootstrap is about 63% of the total sample size,[17 18] which makes its bias (black dashed line) comparable to that of the twofold CV (with 50% of the total sample size; green line). As for the bias of our extrapolation method (red line), it is comparable to that of LOOCV.

In figure 1E, we see that the variance reveals a different story; the LOOCV now has the largest variance, and the twofold CV has the smallest variance among the fold-based CV methods. In terms of variance, the extrapolation method is now comparable to the twofold CV. From the RMSE index (figure 1I), we see that the proposed extrapolation method strikes a good balance between the bias and variance.

Similar results can be found when the variables are correlated (panels B, F and J, with a correlation coefficient of 0.2; panels C, G and K, with a correlation coefficient of 0.5) and when they are not normally distributed (panels D, H and L), or when SVM (figure 2) is used for constructing prediction models.

We simulated a more substantially non-normal data set (see online supplementary appendix 3). We found that the proposed extrapolation method can still strike a good balance between bias and variance (see online supplementary appendix 4). In addition, we examined the performances of the 0.632 bootstrap[17] and the 0.632+ bootstrap,[31] both of which are weighted averages between the LOO bootstrap estimate and the resubstitution estimate. (The 0.632+ bootstrap is an improved version of the 0.632 bootstrap.) We found that the 0.632 bootstrap produces very large upward biases while the 0.632+ bootstrap is quite comparable to our method (see online supplementary appendix 5). We also see that the proposed extrapolation method can outperform the

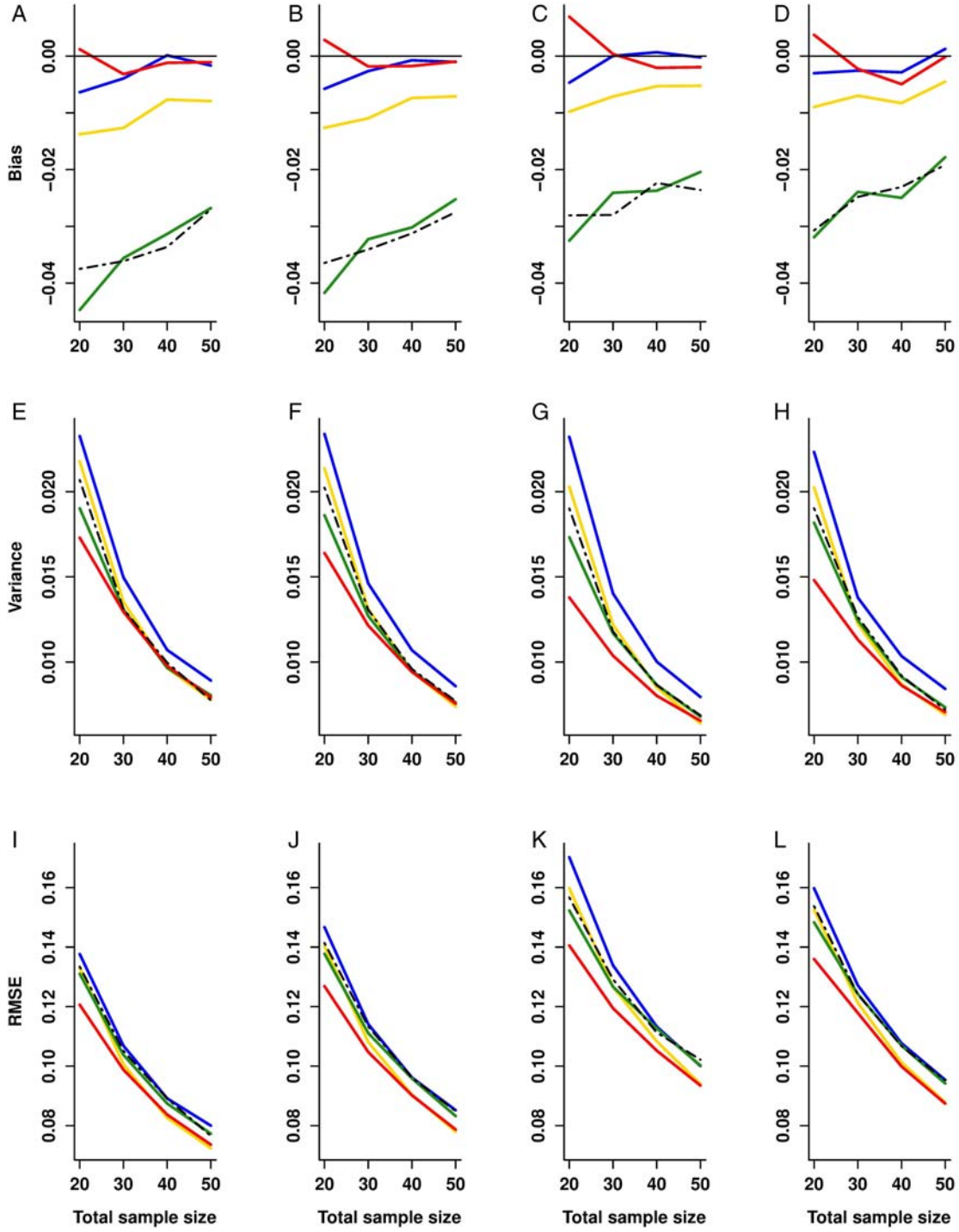


Figure 1 Bias, variance and root mean squared error (RMSE) of the various methods under different sample sizes when the naïve multiple regression is used to build the gene signature leave-one-out cross-validation (blue line), fivefold cross-validation (yellow line), twofold cross-validation (green line), leave-one-out bootstrap (black dashed line) and the proposed method (red line). The leftmost column of panels is for normally distributed data with a correlation coefficient of 0, the second column from left with a correlation coefficient of 0.2, and the third column from left with a correlation coefficient of 0.5. The rightmost column of panels is for complex data (mixture of normal distributions). The horizontal thin lines indicate a position of no bias.

0.632+ bootstrap in terms of RMSE when sample size $< (10 \text{ cases} + 10 \text{ controls})$ (online supplementary appendix 6).

We also tried extrapolation based on different learning curves (a linear equation $y = a + bx$ with $y = \overline{AUC}$ and $x = n_1 + n_0$, and a quadratic equation $y = a + bx + cx^2$ with $y = Z_{\overline{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$), but we found the results to be no better than using the learning curve in this paper (see online supplementary appendices 7 and 8).

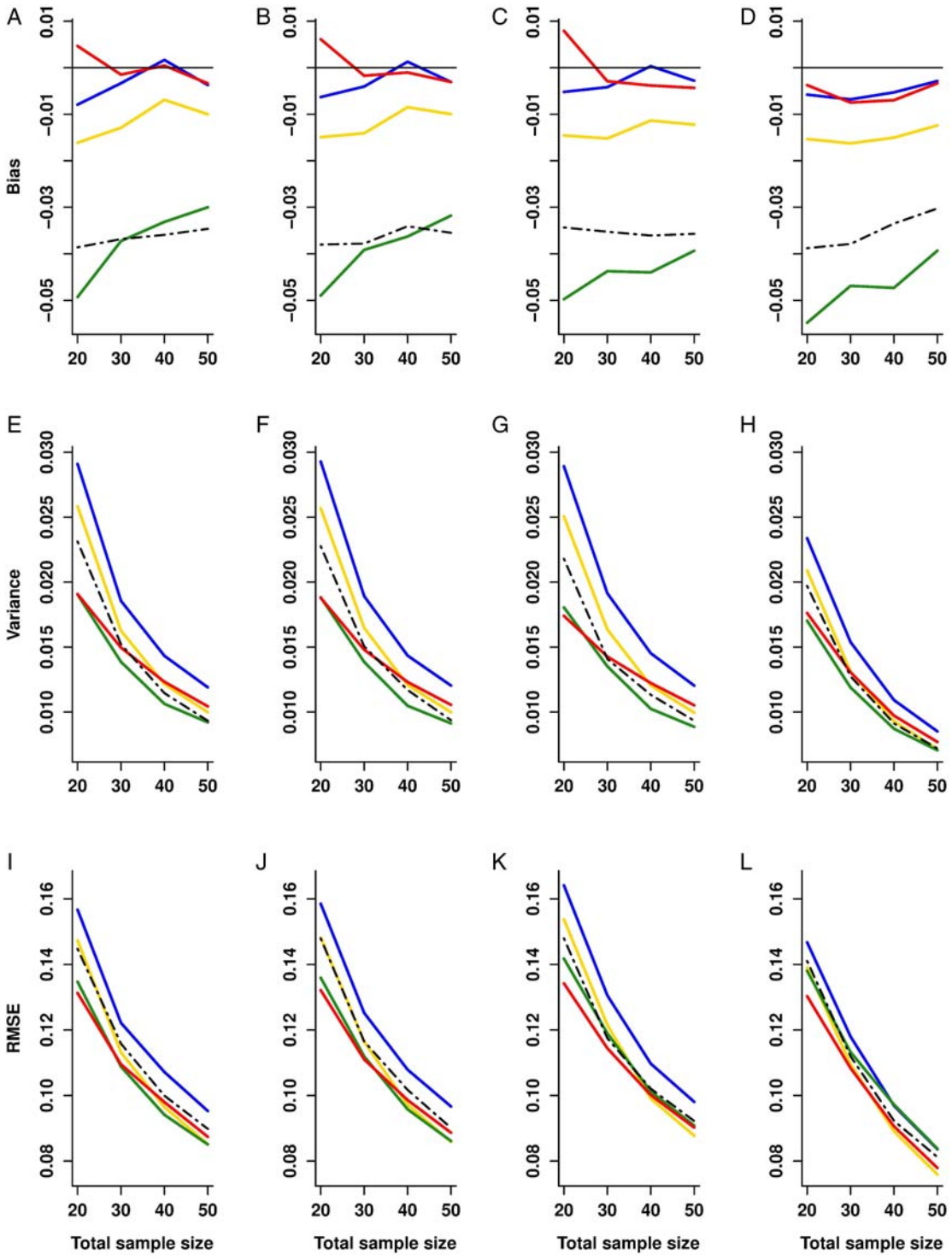


Figure 2 Bias, variance and root mean squared error (RMSE) of the various methods under different sample sizes when the support vector machine is used to build the gene signature leave-one-out cross-validation (blue line), fivefold cross-validation (yellow line), twofold cross-validation (green line), leave-one-out bootstrap (black dashed line) and the proposed method (red line). The leftmost column of panels is for normally distributed data with a correlation coefficient of 0, the second column from left with a correlation coefficient of 0.2, and the third column from left with a correlation coefficient of 0.5. The rightmost column of panels is for complex data (mixture of normal distributions). The horizontal thin lines indicate a position of no bias.

REAL DATA DEMONSTRATION

We take three microarray data sets to demonstrate how the extrapolation method can be applied step by step.[32–34] As this paper focuses on prediction model evaluation, and not on gene selection, we conveniently construct a 10-gene signature based on the top 10 genes with the smallest Mann-Whitney U test p values for the first two data sets, respectively. In the last example, we

directly take the 76-gene signature identified by the original study as the prediction model. Both naïve multiple regression and SVM are used to build the prediction model for each data set. Monte Carlo random partition (a total of 1000 partitions for each fold) is performed to obtain the cross-validated AUCs.

Example 1

The first data set is colon cancer data.[32] The data consist of 2000 gene expressions in 62 tissue samples (40 tumour and 22 normal colon tissue samples). The data are available at http://genomics-pubs.princeton.edu/oncology/. The gene expression level is presented in intensity value, and is otherwise unprocessed. Hence, we first normalise the data by the mean and SD of each gene. The data are then randomly divided into two parts, one for gene selection (28 tumour tissue samples and 10 normal colon tissue samples) and the other for model building and CV (12 tumour tissue samples and 12 normal colon tissue samples). In the gene selection data set, we use the Mann-Whitney U test to identify the top 10 genes with the smallest p value from among the 2000 genes. These are *Hsa.627_M26383*, *Hsa.6814_H08393*, *Hsa.37937_R87126*, *Hsa.692_M76378-3*, *Hsa.3016_T47377*, *Hsa.31630_R64115*, *Hsa.831_M22382*, *Hsa.36689_Z50753*, *Hsa.3331_T86473* and *Hsa.43279_H64489*. In the remaining data set, we build and cross-validate a prediction model for this 10-gene signature.

For naïve multiple regression, the $\overline{\text{AUC}}$s (averaged from 1000 Monte Carlo partitions) are 0.936 (LOOCV), 0.929 (10-fold CV), 0.928 (5-fold CV), 0.925 (3-fold CV) and 0.921 (2-fold CV), respectively. The (x, y)s are then calculated as: $(11^{-1}+11^{-1}, Z^{-2}_{0.936})$=(0.182, 0.432) for LOOCV, (0.200, 0.465) for 10-fold CV, (0.220, 0.466) for 5-fold CV, (0.250, 0.483) for 3-fold CV and (0.330, 0.503) for 2-fold CV, respectively. These results are plotted in figure 3A. We then draw a linear regression based on the five (x, y) points: $y = 0.373 + 0.409x$ (the red line in figure 3A). To predict the performance with a sample size of 24 (all samples in the model building and CV data set are used as the training set, ie, 12 tumour and 12 normal tissue samples), we enter $x = 12^{-1}+12^{-1}$ into the regression equation to get $\hat{y} = 0.441$ (* in figure 3A). The extrapolated performance is therefore $\widehat{\text{AUC}}_T = \Phi\left(\sqrt{0.441^{-1}}\right) = 0.930$. We next perform a total of 100 bootstrapping for this example and the bootstrapped SE for $\widehat{\text{AUC}}_T$ is calculated as 0.080. The results for this example when SVM is used for constructing the prediction model are shown in figure 3B. The $\widehat{\text{AUC}}_T$ (± bootstrapped SE) is calculated as 0.940 (±0.079).

Example 2

The second example is a breast cancer data set,[33] which is available at the Gene Expression Omnibus database (http://www.ncbi.nlm.nih.gov/geo), with accession code GSE2990. These data consist of 22 215 genes for 189 patients with breast cancer (120 patients without relapse and 67 with relapse; 2 patients with unknown relapse status are omitted from our demonstration). The provided data set has already been processed (with background correction, quantile normalisation and log transformation). The data set details and patient profile can be found in the corresponding reference and aforementioned GEO website. We choose those 'extreme' patients to be in the gene selection data set, that is, those 43 patients who developed relapse within 5 years and those 91 patients who were free of relapse for at least 5 years. The remaining data set (for model building and CV) now consists of 67−43=24 patients who developed relapses after 5 years and 120−91=29 patients free of relapses during their less than 5-year follow-up periods.

In the gene selection data set, we again pick the top 10 genes from among the 22 215 genes with the smallest p value after Mann-Whitney U test. These 10 genes are *203213_at*, *210222_s_at*, *205898_at*, *218883_s_at*, *203485_at*, *201890_at*, *214710_s_at*, *202779_s_at*, *202503_s_at* and *201291_s_at*. The remaining data set is used to build and cross-validate the prediction model for this 10-gene signature.

The results for naïve multiple regression are presented in figure 3A (blue line). The one-step extrapolated AUC (± bootstrappedSE) is calculated as 0.803 (±0.082). The results for SVM are presented in figure 3B (blue line). The one-step extrapolated AUC (± bootstrappedSE) is calculated as 0.781 (±0.063).

Example 3

The third example is a different breast cancer data set. The data set[34] and its patient profile are available at the GEO database (http://www.ncbi.nlm.nih.gov/geo) with accession code GSE2034. The data consisting of 107 patients with breast cancer with distant relapse and 197 without distant relapse) was divided into training (115 patients) and testing (171 patients) by concentration of the oestrogen receptor and a 76-gene signature was identified by previous researchers.[34] We use our method to estimate the prediction performance of this 76-gene signature.

The results for naïve multiple regression are presented in figure 3A (green line). The one-step extrapolated AUC (± bootstrappedSE) is calculated as 0.726 (±0.005). The results for SVM are presented in figure 3B (green line). The one-step extrapolated AUC (± bootstrapped SE is calculated as 0.716 (±0.010).

DISCUSSION

When estimating the performance of a model derived from a small study, there seems to be no reason to settle for a sample size of $N_T - 1$ (or $N_T - 2$, in a leave-one-pair-out CV), since what we are looking for is the performance at sample size N_T. In this study, we extrapolate the performance to N_T by exploiting the linear relation between Z^{-2}_{AUC} and $n_1^{-1} + n_0^{-1}$. The extrapolation is

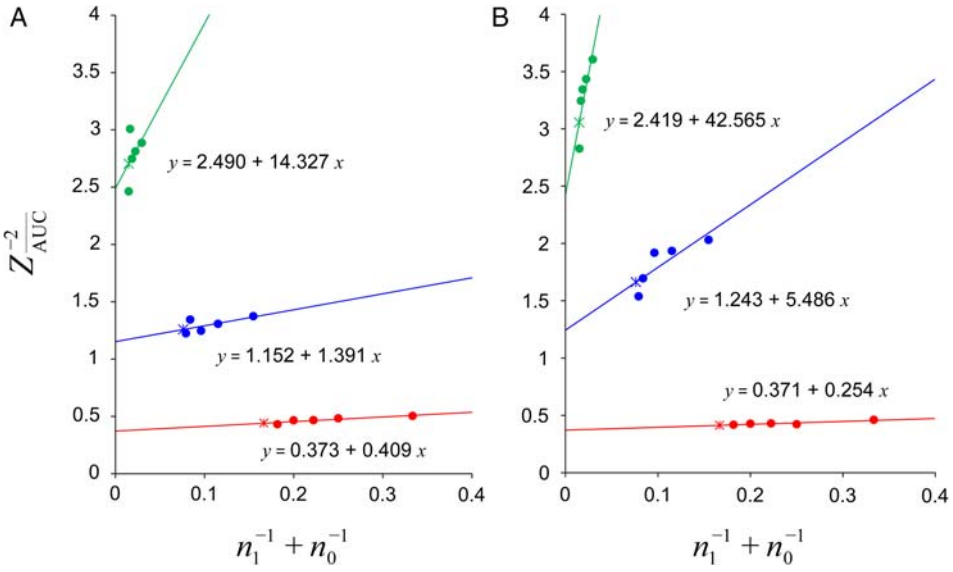


Figure 3 Demonstration of the three microarray data using the proposed extrapolation method. In this double-inverse coordinate system, the ordinate is the inverse of the (transformed) AUC value, and the abscissa is the inverse of the training sample size. The red, blue and green lines represent the colon cancer data (example 1), the GSE2990 breast cancer data (example 2) and the GSE2034 breast cancer data (example 3), respectively. The five dots along each line are the estimates of the five different fold CV and the * is the extrapolated AUC (AUC, area under the receiver operating characteristic curve; CV, cross-validation).

based on five CV methods (LOOCV, 10-fold, 5-fold, 3-fold and 2-fold CV) and is carried out only one-step ahead. The resulting estimate thus inherits the lack of bias in LOOCV and strikes a satisfying variance among the five CV methods. A computer simulation shows that our method performs the best in terms of RMSE when sample sizes are small.

The learning curve for AUC used in this study is based on a linear prediction model (online supplementary appendix 1). However, our simulation study shows that the learning curve is equally suited for non-linear prediction models, such as SVM (figure 2) and RF (online supplementary appendix 9). When making the extrapolation, we may sometimes encounter a slope (b in $y = a + bx$) that is near zero (eg, the colon cancer example demonstrated in figure 3). This may occur when the model performance has reached its plateau; thus, varying the training size (as in different CV methods) has little effect on AUC estimates. This can also occur at the other extreme when the prediction/classification problem at hand is more complex and requires a much larger training size than is currently available, to significantly enhance the model performance. In either case, our method amounts to taking the average of the five CV estimates, thereby stabilising the variances.

Two previous studies[30 35] also exploited the extrapolation concept. Both used an inverse power-law model as the empirical learning curve. In this paper, we are only interested in how the performance will result if all the samples we have are utilised to train the model. We do need an equation (a learning curve) for extrapolation, but this requires extrapolating a mere one step ahead. Our results should therefore be less dependent on what learning curves are being used.

Acknowledgements The authors would like to thank Dr Yung-Hsiang Huang and Mr Po-Chang Hsiao for technical supports.

Contributors L-YW designed the simulation study and drafted the manuscript. W-CL conceived the study and participated in its design and coordination.

Funding Ministry of Science and Technology, Taiwan (grant number NSC 102-2628-B-002-036-MY3) and National Taiwan University, Taiwan (grant number NTU-CESRP-104R7622-8).

Competing interests None declared.

Provenance and peer review Not commissioned; externally peer reviewed.

Data sharing statement No additional data are available.

REFERENCES

1. Cai YD, Huang T, Feng KY, *et al.* A unified 35-gene signature for both subtype classification and survival prediction in diffuse large B-cell lymphomas. *PLoS ONE* 2010;5:e12726.
2. Chibon F, Lagarde P, Salas S, *et al.* Validated prediction of clinical outcome in sarcomas and multiple types of cancer on the basis of a gene expression signature related to genome complexity. *Nat Med* 2010;16:781–7.
3. Levan K, Partheen K, Osterberg L, *et al.* Identification of a gene expression signature for survival prediction in type I endometrial carcinoma. *Gene Expr* 2010;14:361–70.
4. Roessler S, Jia HL, Budhu A, *et al.* A unique metastasis gene signature enables prediction of tumor relapse in early-stage hepatocellular carcinoma patients. *Cancer Res* 2010;70:10202–12.
5. Wan YW, Sabbagh E, Raese R, *et al.* Hybrid models identified a 12-gene signature for lung cancer prognosis and chemoresponse prediction. *PLoS ONE* 2010;5:e12222.
6. Zhu CQ, Ding K, Strumpf D, *et al.* Prognostic and predictive gene signature for adjuvant chemotherapy in resected non-small-cell lung cancer. *J Clin Oncol* 2010;28:4417–24.
7. Chen DT, Hsu YL, Fulp WJ, *et al.* Prognostic and predictive value of a malignancy-risk gene signature in early-stage non-small cell lung cancer. *J Natl Cancer Inst* 2011;103:1859–70.
8. Herold T, Jurinovic V, Metzeler KH, *et al.* An eight-gene expression signature for the prediction of survival and time to treatment in chronic lymphocytic leukemia. *Leukemia* 2011;25:1639–45.
9. Minguez B, Hoshida Y, Villanueva A, *et al.* Gene-expression signature of vascular invasion in hepatocellular carcinoma. *J Hepatol* 2011;55:1325–31.

10. Salazar R, Roepman P, Capella G, *et al*. Gene expression signature to improve prognosis prediction of stage II and III colorectal cancer. *J Clin Oncol* 2011;29:17–24.
11. Wang DY, Done SJ, McCready DR, *et al*. A new gene expression signature, the ClinicoMolecular Triad Classification, may improve prediction and prognostication of breast cancer at the time of diagnosis. *Breast Cancer Res* 2011;13:R92.
12. Xie Y, Xiao G, Coombes KR, *et al*. Robust gene expression signature from formalin-fixed paraffin-embedded samples predicts prognosis of non-small-cell lung cancer patients. *Clin Cancer Res* 2011;17:5705–14.
13. Riester M, Taylor JM, Feifer A, *et al*. Combination of a novel gene expression signature with a clinical nomogram improves the prediction of survival in high-risk bladder cancer. *Clin Cancer Res* 2012;18:1323–33.
14. Schramm SJ, Campain AE, Scolyer RA, *et al*. Review and cross-validation of gene expression signatures and melanoma prognosis. *J Invest Dermatol* 2012;132:274–83.
15. Simon R. Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *Br J Cancer* 2003;89:1599–604.
16. Molinaro AM, Simon R, Pfeiffer RM. Prediction error estimation: a comparison of resampling methods. *Bioinformatics* 2005;21:3301–7.
17. Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation. *J Am Stat Assoc* 1983;78:316–31.
18. Efron B, Tibshirani RJ. *An introduction to the bootstrap*. CRC Press, 1994.
19. Refaeilzadeh P, Tang L, Liu H. *Cross-validation*. Encyclopedia of database systems. Springer, 2009:532–8.
20. Dougherty ER. Small sample issues for microarray-based classification. *Comp Funct Genomics* 2001;2:28–34.
21. Braga-Neto UM, Dougherty ER. Is cross-validation valid for small-sample microarray classification? *Bioinformatics* 2004;20:374–80.
22. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 1982;143:29–36.
23. Parker BJ, Günter S, Bedo J. Stratification bias in low signal microarray studies. *BMC Bioinform* 2007;8:326.
24. Airola A, Pahikkala T, Waegeman W, *et al*. A comparison of AUC estimators in small-sample studies. *3rd International workshop on Machine Learning in Systems Biology (MLSB 09)* 2009:15–23.
25. Yelle LE. The learning curve: Historical review and comprehensive survey. *Decis Sci* 1979;10:302–28.
26. Radmacher MD, McShane LM, Simon R. A paradigm for class prediction using gene expression profiles. *J Comput Biol* 2002;9:505–11.
27. Vapnik VN. An overview of statistical learning theory. *IEEE Trans Neural Netw* 1999;10:988–99.
28. Karatzoglou A, Meyer D, Hornik K. Support vector machines in R. *J Stat Softw* 2006;15:1–28.
29. Breiman L. *Out-of-bag estimation*. Technical report. Berkeley, CA: Department of Statistics, University of California, 1996.
30. Jiang W, Simon R. A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification. *Stat Med* 2007;26:5320–34.
31. Efron B, Tibshirani R. Improvements on cross-validation: the 632 +bootstrap method. *J Am Statist Assoc* 1997;92:548–60.
32. Alon U, Barkai N, Notterman DA, *et al*. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA* 1999;96:6745–50.
33. Sotiriou C, Wirapati P, Loi S, *et al*. Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. *J Natl Cancer Inst* 2006;98:262–72.
34. Wang Y, Klijn JG, Zhang Y, *et al*. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet* 2005;365:671–9.
35. Mukherjee S, Tamayo P, Rogers S, *et al*. Estimating dataset size requirements for classifying DNA microarray data. *J Comput Biol* 2003;10:119–42.

Supplemental Materials

Appendix 1

Suppose that we have a total of N_1 cases, and a total of N_0 controls. A gene signature containing a total of p genes (indexed by $i = 1,..., p$) is used to predict a binary outcome. The training samples are used to construct the simple linear regression with the beta coefficients (β_i) calculated as the mean difference of each variable between the case and the control groups. We let random variable $\chi_{i,1}$ ($\chi_{i,0}$) denote the gene expression level of the ith gene for a randomly selected case (control) in the testing set. Assume that gene expression levels are normally distributed: $\chi_{i,1} \sim N(\mu_{i,1}, \sigma^2)$ and $\chi_{i,0} \sim N(\mu_{i,0}, \sigma^2)$. The effect size for the ith gene is $\tau_i = \left| \frac{\mu_{i,1} - \mu_{i,0}}{\sigma} \right|$, which corresponds to an AUC (area under the receiver operating characteristic curve) of $\text{AUC}_i = \Phi(\tau_i / \sqrt{2})$ [1,2]. Let $D = \sum_{i=1}^{p} \hat{\beta}_i \chi_{i,1} - \sum_{i=1}^{p} \hat{\beta}_i \chi_{i,0}$ be the random variable representing the difference in risk scores between case and control in a randomly selected pair in the testing set.

The expectation and the variance of D for an average set of $\hat{\beta}_i$ coefficients derived from the training sample are $\text{E}(D) = \sum_{i=1}^{p} (\mu_{i,1} - \mu_{i,0})^2$ and $\text{Var}(D) = 2\sigma^2 \sum_{i=1}^{p} (\mu_{i,1} - \mu_{i,0})^2 + 2p\sigma^4 \left(n_1^{-1} + n_0^{-1} \right)$. Therefore for each fold, the AUC for the gene signature using these *p* genes, as evaluated by the testing sample is,

$$\text{AUC} = 1 - \Phi\left(\frac{0 - \text{E}(D)}{\sqrt{\text{Var}(D)}} \right) = \Phi\left(\bar{\tau} \times \sqrt{\frac{p}{2} \cdot \left(1 + \left(n_1 \bar{\tau}^2 \right)^{-1} + \left(n_0 \bar{\tau}^2 \right)^{-1} \right)^{-1}} \right),$$ where

$\bar{\tau} = \sqrt{\frac{1}{p} \sum_{i=1}^{p} \tau_i^2}$ is the average (root mean square) effect size of p genes. This AUC equation can be transformed into a simple linear equation: $Z_{\text{AUC}}^{-2} = \frac{2}{p\bar{\tau}^2} + \frac{2}{p\bar{\tau}^4} \times (n_1^{-1} + n_0^{-1})$. This is standard linear regression line which can be presented in the form of $y = a + bx$. Therefore, we use $y = Z_{\text{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$

in the text for linear extrapolation.

For the learning curve, we expect $b > 0$ because AUC should increase with the training sample size. We also expect $a > 0$ because Z_{AUC}^{-2} should be a positive value. Sometimes, the regression equation has $b \le 0$. When this occurs, we let $Z_{\text{AUC}_\text{T}}^{-2}$ be the average of the above five Z_{AUC}^{-2}s, and if $a \le 0$, we perform a linear regression without the intercept, i.e., $y = bx$ and let $Z_{\text{AUC}_\text{T}}^{-2} = b \times (N_1^{-1} + N_0^{-1})$.

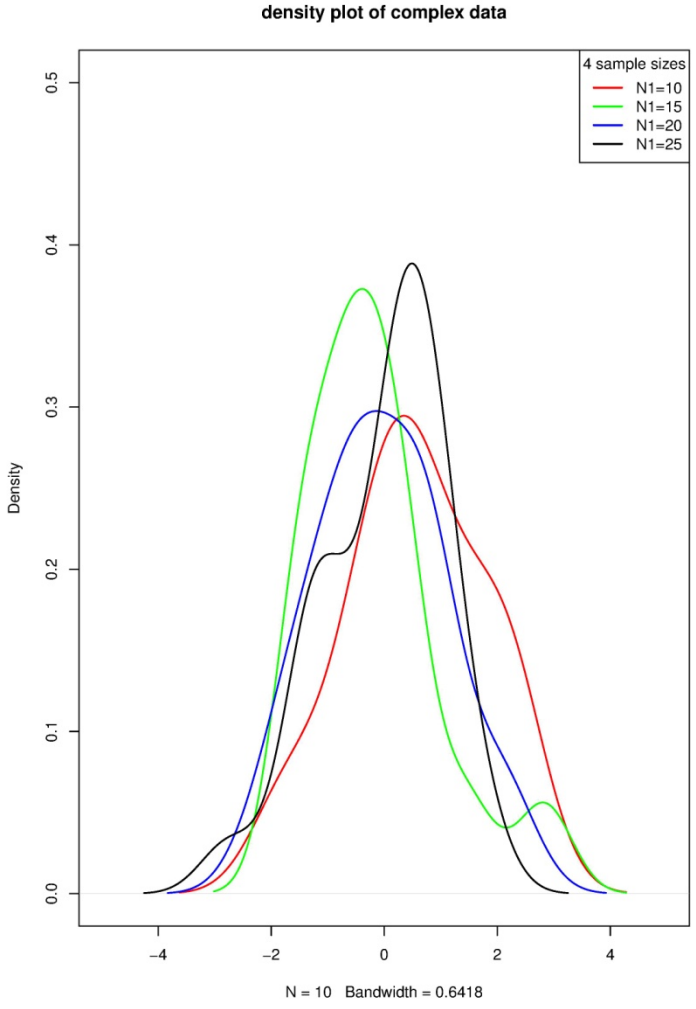
References

1. Wolfe D HR. On constructing statistics and reporting data. *Am Stat*. 1971;25(4):27-30.
2. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 1982;143(1):29-36.

.

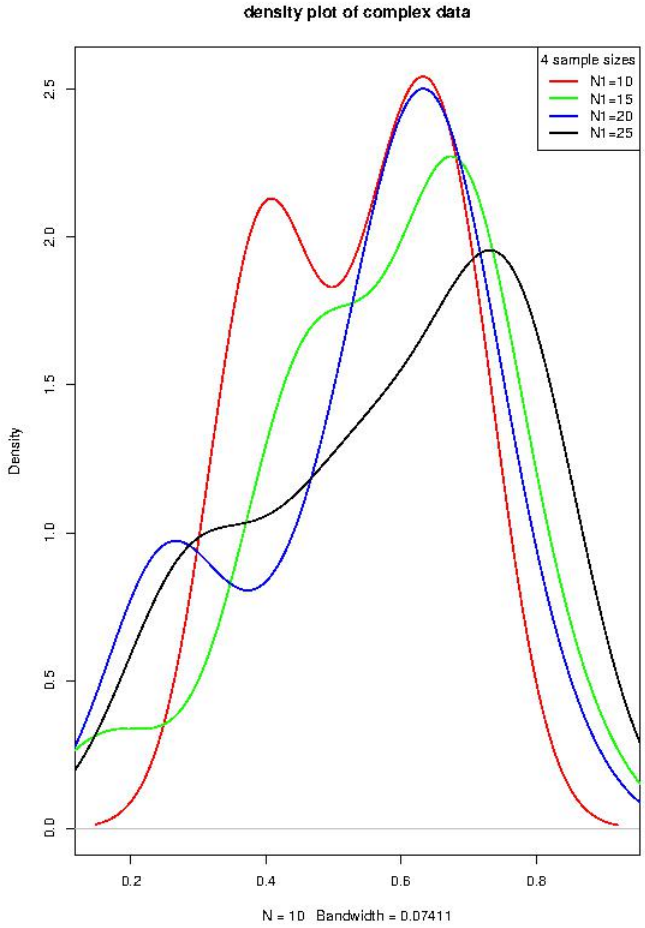
Appendix 2

The more complex data is a mixture of three normal distributions. For the cases, the expression level of each gene is distributed as a mixture of three normal distributions with variances of 1. The three means are generated from a uniform [-1.5, 1.5] distribution with a probability of 0.6, a uniform [-1.2, 1.2] distribution with a probability of 0.3 and a uniform [-1, 1] distribution with a probability of 0.1, respectively. The following displays this normal-mixture distribution (in one simulation). Different colors represent the four sample sizes considered in our simulation studies (distribution of controls not shown).



Appendix 3

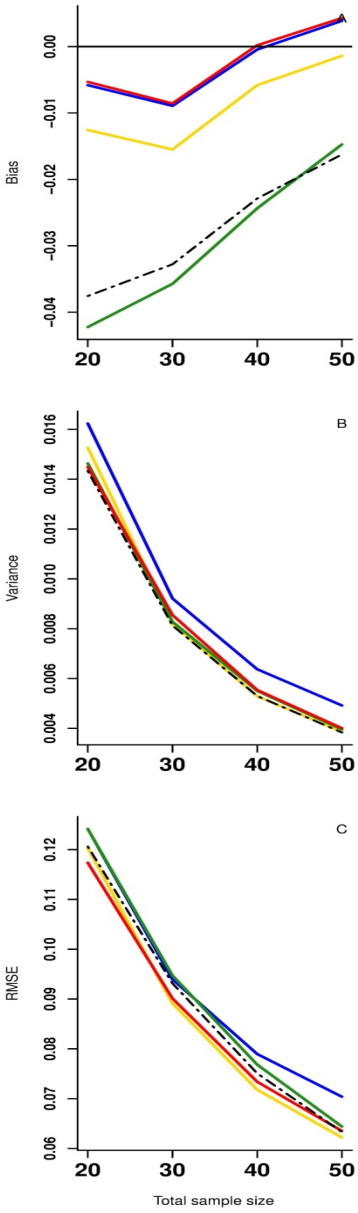
The substantially non-normal data is a mixture of two beta distributions with equal probability for both cases and controls. For the cases, (α, β)=(5,6) and (10,5) for the two beta distributions, both with non-centrality parameter 0. For the controls, (α, β) =(4,3) and (2,1) for the two beta distributions, both with non-centrality parameter 0. The following displays this beta-mixture distribution (in one simulation). Different colors represent the four sample sizes considered in our simulation studies (distribution of controls not shown).



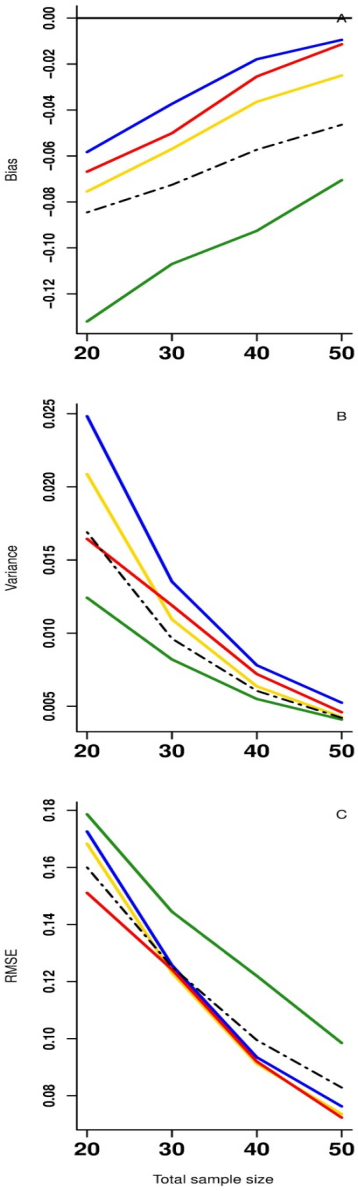
Appendix 4

Simulation results for the beta-mixture distribution [leave-one-out cross validation (blue line), 5-fold cross validation (yellow line), 2-fold cross validation (green line), leave-one-out bootstrap (black dashed line), and the proposed method (red line).]

(1) Naive multiple regression



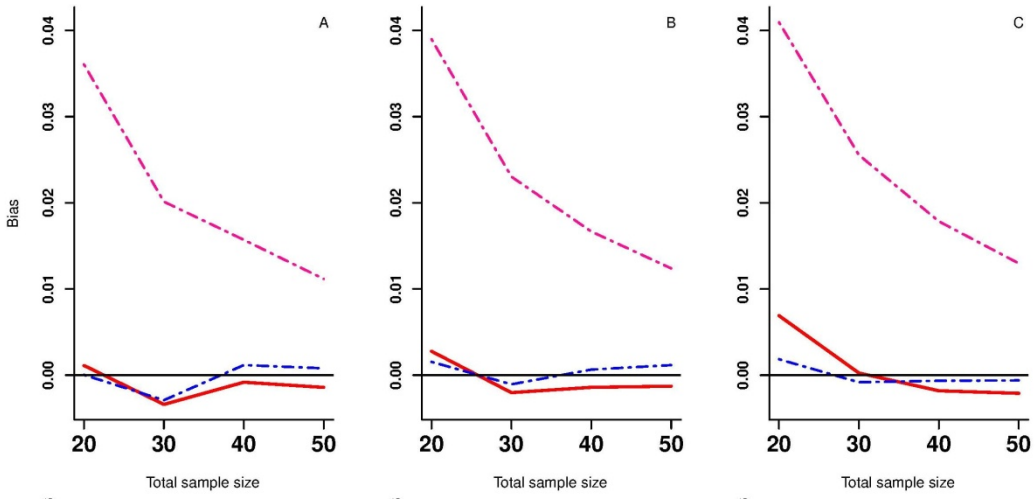
(2) Support vector machine



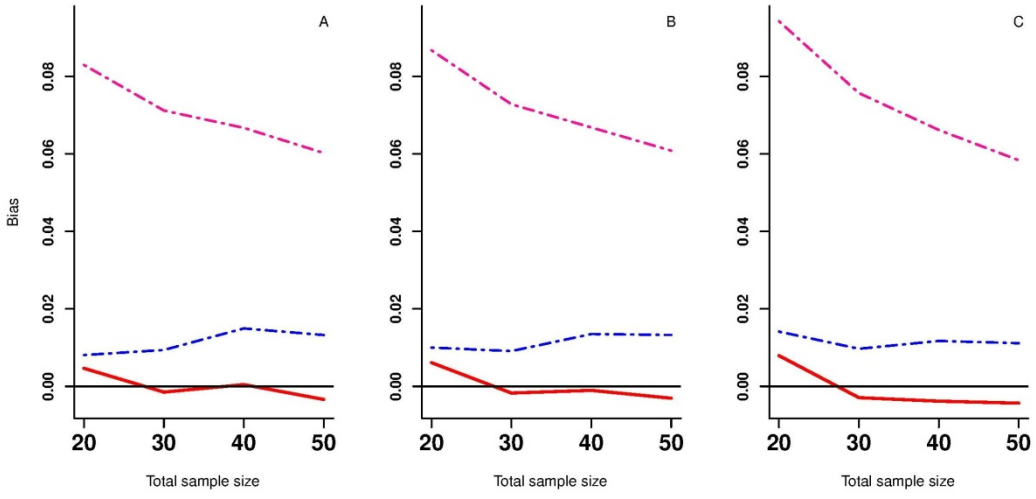
Appendix 5

The 0.632 bootstrap (pink dashed line) shows large bias while the bias of the 0.632+ bootstrap (blue dashed line) is comparable to the proposed method (red line) when naive multiple regression and support vector machine are used to build gene signature. The panel A is for normally distributed data with correlation coefficient of 0, the panel B, with correlation coefficient of 0.2, the panel C is with correlation coefficient of 0.5. The horizontal thin lines indicate the position of no bias.

(1) Naive multiple regression



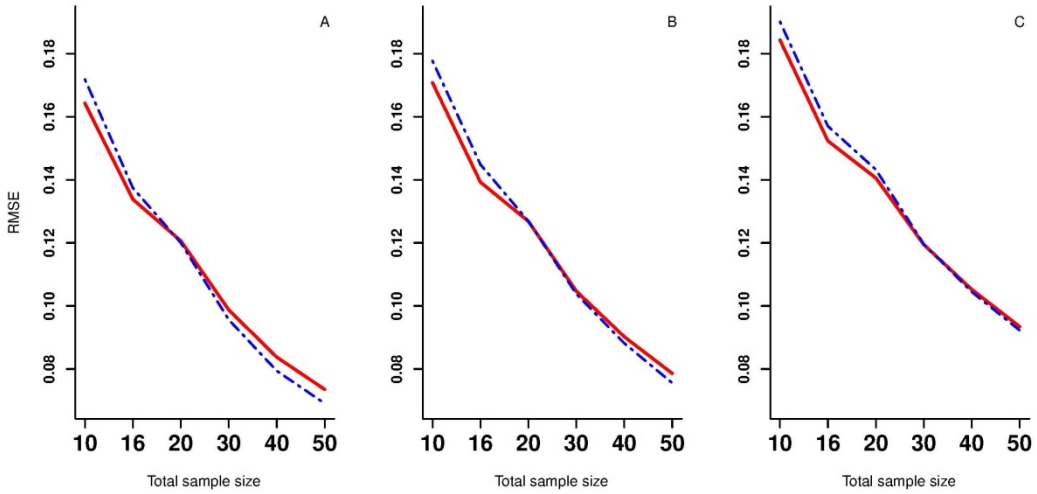
(2) Support vector machine



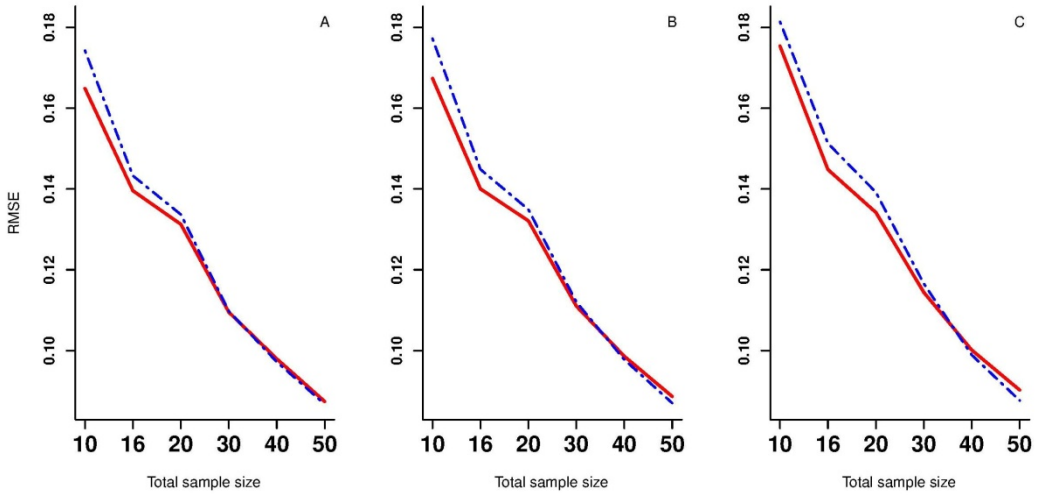
Appendix 6

The proposed method (red line) outperforms the 0.632+ bootstrap (blue dashed line) in terms of RMSE when sample size $< (10 \text{ cases} + 10 \text{ controls})$ when naive multiple regression and support vector machine are used to build gene signature. The panel A is for normally distributed data with correlation coefficient of 0, the panel B, with correlation coefficient of 0.2, the panel C is with correlation coefficient of 0.5.

(1) Naive multiple regression

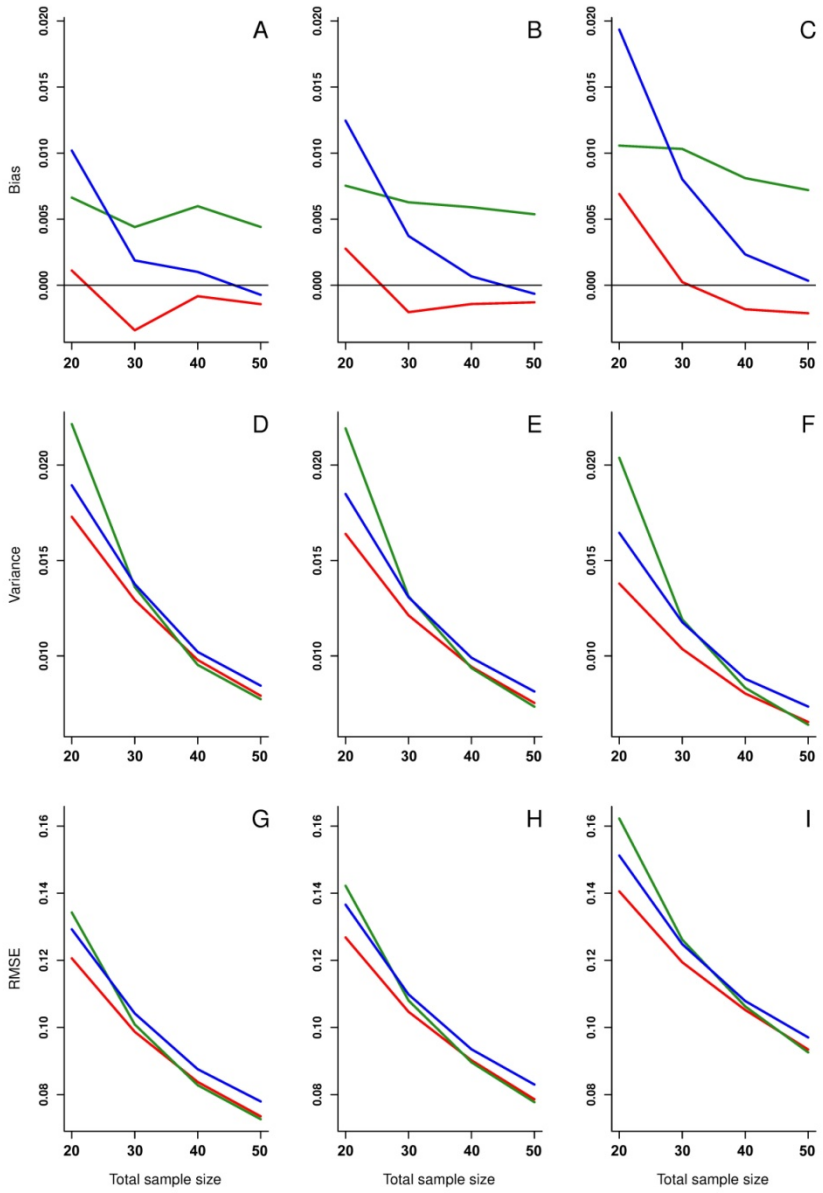


(2) Support vector machine



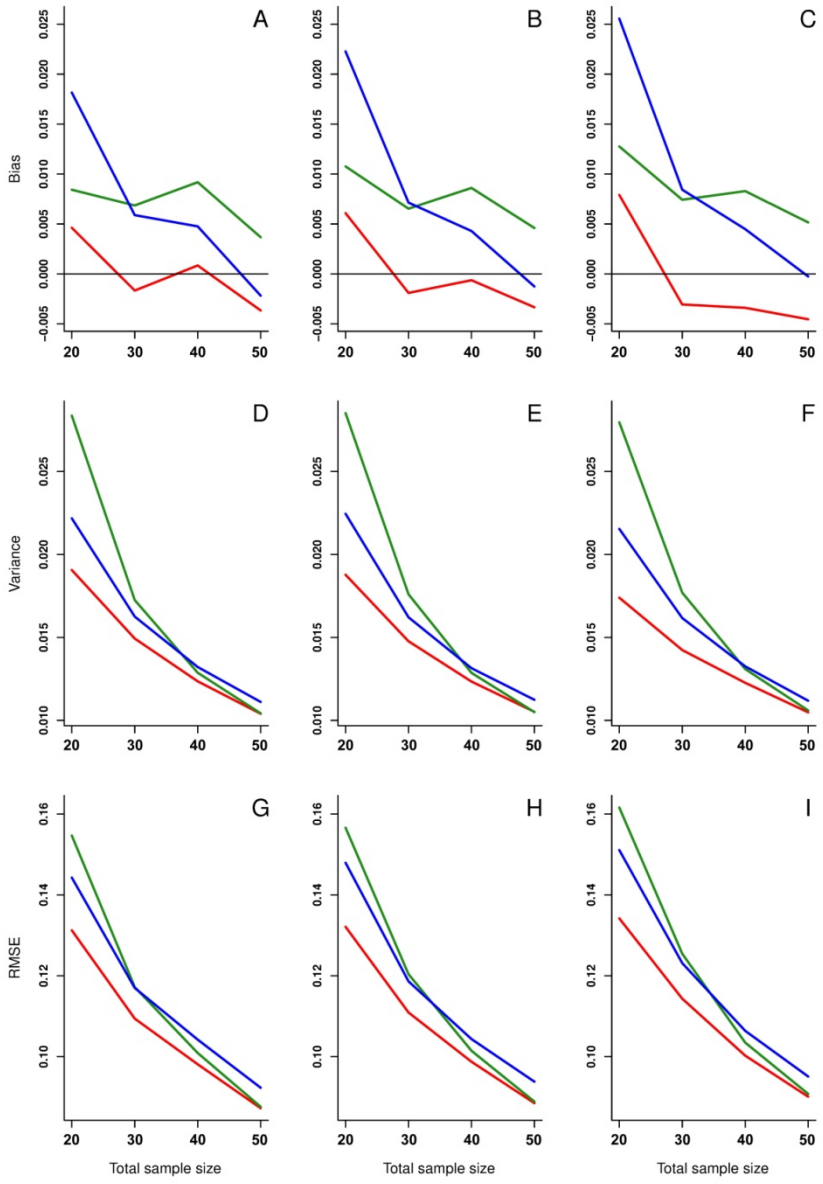
Appendix 7

Comparison of bias, variance, and root mean squared error (RMSE) of the extrapolation methods using different learning curves : $y = a + bx$ with $y = Z_{\mathrm{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$ (red, learning curve used in this study), $y = a + bx$ with $y = \overline{\mathrm{AUC}}$ and $x = n_1 + n_0$ (green) , $y = a + bx + cx^2$ with $y = Z_{\mathrm{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$ (blue) when the naive multiple regression is used to build the gene signature. The left column of panels is for normally distributed data with correlation coefficient of 0, the middle column is correlation coefficient of 0.2, and the right column is correlation coefficient of 0.5. The horizontal thin lines indicate the position of no bias.



Appendix 8

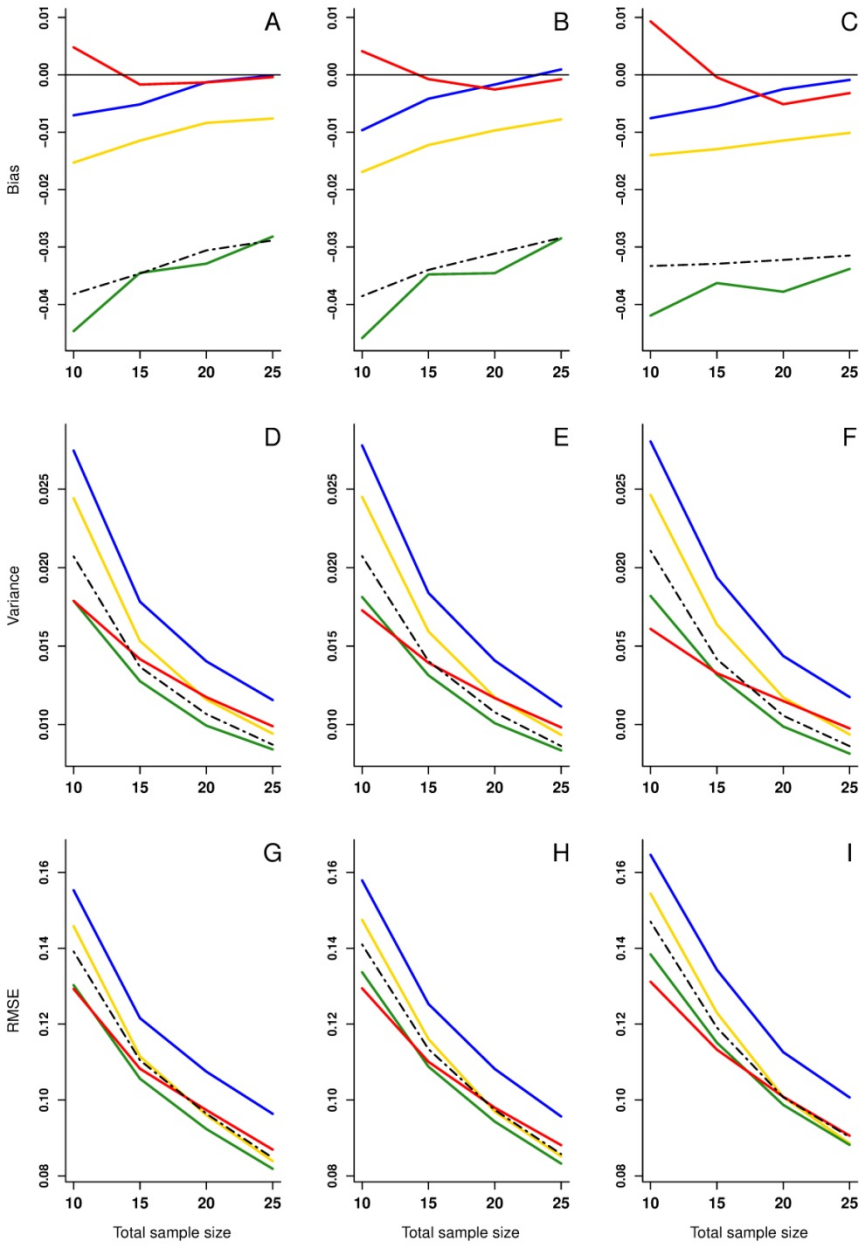
Comparison of bias, variance, and root mean squared error (RMSE) of the extrapolation methods using different learning curves : $y = a + bx$ with $y = Z_{\mathrm{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$ (red, learning curve used in this study), $y = a + bx$ with $y = \overline{\mathrm{AUC}}$ and $x = n_1 + n_0$ (green) , $y = a + bx + cx^2$ with $y = Z_{\mathrm{AUC}}^{-2}$ and $x = n_1^{-1} + n_0^{-1}$ (blue) when the support vector machine is used to build the gene signature. The left column of panels is for normally distributed data with correlation coefficient of 0, the middle column is correlation coefficient of 0.2, and the right column is correlation coefficient of 0.5. The horizontal thin lines indicate the position of no bias.



Appendix 9

Bias, variance, and root mean squared error (RMSE) of the various methods under different sample size when the random forest is used to build the gene signature [leave-one-out cross validation (blue line), 5-fold cross validation (yellow line), 2-fold cross validation (green line), leave-one-out bootstrap (black dashed line), and the proposed method (red line). The leftmost column of panels is for normally distributed data with correlation coefficient of 0, the second column from left, correlation coefficient of 0.2, and the third column from left, correlation coefficient of 0.5. The horizontal thin lines indicate the position of no bias.

(Random Forest is another machine learning algorithm fisrt introduced by Breiman[1]. It is an ensemble of unpruned classification or regression trees generated by using bootstrap samples of the training data and random feature selection in tree induction. The trees then vote for the most popular class of the ensemble. In this study, we use the randomForest - package of R with ntree (number of trees) 500 and mtry (munber of variables randomly sampled as candidates at each split) 3.).



References

1. Breiman L. Random forests. *Mach Learn* 2001;45(1):5-32.