BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 22 2011, pages 3129–3134
doi:10.1093/bioinformatics/btr542

Gene expression

Advance Access publication September 28, 2011

Lower confidence bounds for prediction accuracy in high dimensions via AROHIL Monte Carlo

Kevin K. Dobbin[1,*] and Stephanie Cooke[2]

[1]Department of Epidemiology and Biostatistics and [2]Institute of Bioinformatics, University of Georgia, Athens, GA, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Implementation and development of statistical methods for high-dimensional data often require high-dimensional Monte Carlo simulations. Simulations are used to assess performance, evaluate robustness, and in some cases for implementation of algorithms. But simulation in high dimensions is often very complex, cumbersome and slow. As a result, performance evaluations are often limited, robustness minimally investigated and dissemination impeded by implementation challenges. This article presents a method for converting complex, slow high-dimensional Monte Carlo simulations into simpler, faster lower dimensional simulations.

Results: We implement the method by converting a previous Monte Carlo algorithm into this novel Monte Carlo, which we call AROHIL Monte Carlo. AROHIL Monte Carlo is shown to exactly or closely match pure Monte Carlo results in a number of examples. It is shown that computing time can be reduced by several orders of magnitude. The confidence bound method implemented using AROHIL outperforms the pure Monte Carlo method. Finally, the utility of the method is shown by application to a number of real microarray datasets.

Availability: The R computer program for forming confidence bounds is freely available for download at the URL http://dobbinke.myweb.uga.edu/RprogramAROHILloweraccuracybound.txt.

Contact: dobbinke@uga.edu

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on January 11, 2011; revised on June 13, 2011; accepted on September 25, 2011

1 INTRODUCTION

This article presents a novel approach to Monte Carlo simulations in high dimensions. The approach permits routines to be written in simpler and faster code using mathematical modeling. The savings comes from reducing the computational dimension of the Monte Carlo simulation from very high dimension to a much lower dimensional space. As we will show, this computational savings makes it possible to provide R functions to perform statistical analyses that previously required a compiled language such as C++. Moreover, the R programs are significantly faster and more transparent (enhancing reproducibility) than the compiled programs, because the underlying models have been streamlined. We make available such a program with this publication. More generally, this approach, by providing a faster method for performing simulations, can enable method developers to consider the robustness of novel procedures across a wider range of simulation scenarios than would otherwise be feasible.

Monte Carlo simulations are commonly encountered in papers on high-dimensional methodologies. Perhaps the most common use of Monte Carlo simulations is to evaluate the performance characteristics of novel statistical procedures, such as the performance of classifiers based on partial least squares (Nguyen and Rocke, 2004), regularization methods for variable selection (Zou and Hastie, 2005) or evaluation of multiple hypothesis testing error control methods (Dudoit *et al.*, 2003). The advantages of Monte Carlo investigations are that the truth can be known exactly, and model assumptions can be violated in systematic ways to explore the limits of robustness. Some other methodologies also use Monte Carlo simulations as part of their algorithms.

This new simulation procedure is called Adequate Representation Of High dimensions In Low dimensions (AROHIL) Monte Carlo. There are two types of AROHIL Monte Carlo. The first type does not involve any resampling. The basic idea behind this type of Monte Carlo is to split the Monte Carlo simulation into two subsimulations. One simulation represents the dimension-reducing feature selection step. The second simulation represents the conditional distribution of the features given that they were selected, and can typically be carried out in a space with dimension similar to the number of features selected.

The second type of AROHIL Monte Carlo does involve resampling, such as bootstrap or cross-validation. In this case, resampling creates complex inter-relationships among the resampled datasets. To capture these inter-relationships, we propose a relatively simple hierarchical model that requires generation of a single high-dimensional vector, and then a series of low-dimensional vectors conditionally generated given the high-dimensional vector.

To our knowledge, there has not been work to develop a general methodology along the lines presented here. Work with a similar spirit can be seen in the high-dimensional literature. For example, Venkatraman and Olshen (2007) developed a faster version of their earlier method (Olshen *et al.*, 2004) for performing circular binary segmentation. Monte Carlo methods for estimating the distribution of functionals in complex statistical models have a longer history, and include rejection sampling, importance sampling (e.g. Ripley, 1987), Markov chain Monte Carlo (e.g. Robert and Casella, 2004) and related algorithms such as the Gibbs sampler (Geman and Geman, 1984). But these methods do not achieve the reduction in the

[*]To whom correspondence should be addressed.

Table 1. AROHIL Monte Carlo applicability for some simple settings

	Covariance matrix	Σ diagonal		Σ positive definite	
	Selection process	*P*-value cutoff	Top k	*P*-value cutoff	Top k
AROHIL Models	Selection model	Indep. Bernoulli R.V.'s	Fixed number k	MVN and Wishart Model	Fixed number k
	Cond. Distn. model	Indep truncated distn.'s	Top k order statistics	MVN and Wishart trunc. distn. model	MVN and Wishart model, top k order stats.

AROHIL Monte Carlo modeling for homoscedastic multivariate normal data. The predictor development algorithm in each case is the compound covariate predictor (CCP). Settings do not involve resampling. Details are provided in the Supplementary Material.

dimension of the computation space, which is the key computational advantage of AROHIL Monte Carlo.

Section 2 discusses applicability of the method, presents the AROHIL method and uses examples to display the approach. Section 3 presents the results of the simulations and applications to real datasets. Section 4 discusses the conclusions and potential areas for future work.

2 METHODS

AROHIL requires that the high-dimensional model be converted into two submodels: (i) a feature selection model and (ii) a conditional distribution in the reduced-dimensional space model. The feature selection model (i) reflects the transition from a high dimensional space to a low dimensional space. The conditional distribution model (ii) reflects the distribution of low-dimensional statistics conditional on their inclusion in the feature selection model—that is, the distribution induced by the feature selection step. Below, we discuss examples of each for settings in which there is no resampling, and in which resampling is involved. The complexity of modeling either step will vary with the probability model and the prediction algorithm.

2.1 Applicability

The computer program and related working examples presented in this article are complicated implementations of results in Dobbin (2009). The complexity may obscure the range of applicability of the method. Table 1 presents some simpler contexts for AROHIL Monte Carlo. The first row of the table are the feature selection models. The second row of the table are the conditional distribution models. The columns of the table represent different multivariate normal data models. The compound covariate predictor (Radmacher *et al.*, 2002) is the prediction algorithm. Modifications can be made to adjust for different multivariate data distributions and prediction algorithms, although these may be non-trivial in some cases.

The first row of Table 1 presents the selection models. For example, consider the cell with 'Indep. Bernoulli R.V.'s'. It indicates that independent Bernoulli random variables can be used when data are multivariate normal, with diagonal covariance matrix, and genes are selected one-at-a-time based on *t*-test *P*-values being below a threshold (such as 0.001). The success probability is the power if the gene is differentially expressed, or the significance level if the gene is not differentially expressed. On the other hand, as indicated in the cells with 'Fixed number k', if the top *k* features are selected for the model, where *k* is fixed, then the selection 'model' is deterministic and just returns the number *k* always. Finally, in the setting of a more complex covariance structure, the selection model needs further refinement. Nevertheless, even in the most general setting of a positive definite covariance matrix of arbitrary form (cell with 'MVN and Wishart Model'), a single multivariate normal variate and a single Wishart variate are all that are required. Details are provided in the Supplementary Material.

The second row of Table 1 presents the conditional distribution models. For example, consider the cell with 'Indep truncated distn's'. Here, the covariance is diagonal and a *P*-value cutoff is used to select features. Weights for the CCP are *t*-test statistics. Thus, the distribution of the predictor in the reduced space is the same as a set of independent, truncated, non-central Student's *t* distributions. Non-centrality parameters can be calculated from the model parameters. Truncation points are determined by the feature selection stringency used. The distribution of the CCP cutpoint for classification can be generated in a fairly simple way. Details are described in the Supplementary Material.

2.2 Working example

The computer program provided with this article uses AROHIL Monte Carlo for several objectives. When no resampling is required, models are similar to the example in Column 1 of Table 1, and details are given in Section 5 of the Supplementary Material. The discussion below will focus on the more challenging context of high-dimensional resampling. The objective will be to obtain the quantiles of the distribution of the predictive accuracy estimate from leave-one-out cross-validation. This working example of resampling-based AROHIL Monte Carlo is motivated by previous work (Dobbin, 2009).

In our working example, the Monte Carlo simulation data are from the homoscedastic multivariate normal model

$$\begin{aligned} x_i &\sim MVN(\mu_x, D), i=1,...,n/2 \\ y_i &\sim MVN(\mu_y, D), i=1,...,n/2 \end{aligned} \qquad (1)$$

where D is a diagonal covariance matrix. Let $2\delta_j$ be the j-th element of $\mu_x - \mu_y$, and σ_j^2 the j-th diagonal element of D.

Features are selected with pooled variance t-test P-values below a threshold α. Selected features are given weights equal to the pooled variance t-test statistic, and unselected feature are given weight zero. Let $\hat{L}$ be the $p \times 1$ vector of weights. Let w be a $p \times 1$ future observation. The classification rule is

$$\begin{aligned} c &= \frac{1}{2}\left(\hat{L}'\bar{X} + \hat{L}'\bar{Y}\right) \\ \text{CCP}(w) &= C_x 1_{\{(\hat{L}'w-c)(\hat{L}'\bar{X}-c)>0\}} + \\ &\quad C_y 1_{\{(\hat{L}'w-c)(\hat{L}'\bar{X}-c)\leq 0\}} \end{aligned}$$

where 1_A is the indicator function of event A, and C_x is the class of x's and C_y the class of y's.

2.3 AROHIL Monte Carlo models for resampling-based feature selection

The feature selection step in the context of traditional Monte Carlo resampling requires generating data from the probability model $X \sim F$ where X is $n \times p$ (Supplementary Figs S3 and S4). The next step is to resample from X to produce $\tilde{X}$ which is $\tilde{n} \times p$. For example, in LOOCV $\tilde{n} = n-1$; in bootstrap, usually $\tilde{n} = n$. Then $\tilde{T}_{MC} = g(\tilde{X})$ is a vector of statistics used for feature selection, and $\tilde{S}_{MC} = h(\tilde{T}_{MC})$ a binary vector of feature selection indicators. Then, the same dataset is resampled from multiple times, denoted r times, producing $\tilde{S}^1_{MC}, \ldots, \tilde{S}^r_{MC}$; for example, in LOOCV $r = n$; in bootstrap, $r = B$ where B is the number of bootstrap samples.

Ideally, the AROHIL approach will produce a set of indicator vectors $\tilde{S}^1_{\mathrm{AROHIL}}, \ldots, \tilde{S}^r_{\mathrm{AROHIL}}$ such that

$$\begin{bmatrix} \tilde{S}^1_{\mathrm{AROHIL}} \\ \cdots \\ \tilde{S}^r_{\mathrm{AROHIL}} \end{bmatrix} \overset{d}{=} \begin{bmatrix} \tilde{S}^1_{MC} \\ \cdots \\ \tilde{S}^r_{MC} \end{bmatrix}.$$

One way to do this is to generate first

$$\begin{bmatrix} \tilde{T}^1_{\mathrm{AROHIL}} \\ \cdots \\ \tilde{T}^r_{\mathrm{AROHIL}} \end{bmatrix} \overset{d}{=} \begin{bmatrix} \tilde{T}^1_{MC} \\ \cdots \\ \tilde{T}^r_{MC} \end{bmatrix}. \tag{2}$$

Let B_{full} be a vector of statistics for feature selection based on the full dataset. Consider the following hierarchical model,

$$B_{\mathrm{full}} \sim F_B$$
$$\begin{bmatrix} \tilde{T}^1_{\mathrm{AROHIL}} \\ \cdots \\ \tilde{T}^r_{\mathrm{AROHIL}} \end{bmatrix} \Bigg| B_{\mathrm{full}} \sim F_T(\cdot|B)$$

if the distribution functions F_B and $F_T(\cdot|B)$ can be derived from the model, then this can ensure Equation (2) is satisfied.

For the working example, the backbone vector B_{full} is the P-dimensional vector of pooled T statistics from the full dataset. The distribution of the each element of B_{full} is independent: Student's t with $n-2$ degrees of freedom and non-centrality parameter calculated from the model parameters. Therefore, generating B_{full} is straightforward.

Let $\bar{X}_{(i)}$ and $D_{(i)}$ be the mean and pooled covariance estimates when sample i is left out from the X's. It is shown in the Supplementary Material that for large n we have the approximation,

$$\left(\bar{X}_{(i)} - \bar{Y}\right)' \hat{D}^{-1}_{(i)} | B_{\mathrm{full}} \sim \mathrm{MVN}\left(B_{\mathrm{full}}, \frac{4}{n(n-2)}\right).$$

Then the powers can be calculated for feature g by

$$q_g \approx 1 - \Phi\left[\sqrt{n(n-2)}\left(\left[\sqrt{4/n}\right] t_{\alpha/2,n-2} - B_{\mathrm{full},g}\right)\right] + \Phi\left[\sqrt{n(n-2)}\left(-\left[\sqrt{4/n}\right] t_{\alpha/2,n-2} - B_{\mathrm{full},g}\right)\right]$$

where $B_{\mathrm{full},g}$ is the g-th element of B_{full}. Finally, each $S^r_{\mathrm{AROHIL}}|B_{\mathrm{full}}$ is a vector of independent (conditional on B_{full}) Bernoulli random variables, where the g-th entry has success probability q_g.

2.4 AROHIL Monte Carlo models for resampling-based conditional distributions

After gene selection, traditional Monte Carlo works in the reduced dimensional space. Let $k(r)$ be the dimension of the reduced space on resampled dataset r, and $T_{MC,k(r)}$ the vector of statistics in the reduced dimensional space. The objective function of interest is $Q_{MC,k(r)} = h(T_{MC,k(r)})$. AROHIL Monte Carlo generates $T_{\mathrm{AROHIL},k(r)}$ directly from a probability model. Then $Q_{\mathrm{AROHIL},k(r)} = h(T_{\mathrm{AROHIL},k(r)})$ and ideally one wants,

$$\begin{pmatrix} Q^1_{\mathrm{AROHIL},k(1)} \\ \cdots \\ Q^r_{\mathrm{AROHIL},k(r)} \end{pmatrix} \overset{d}{=} \begin{pmatrix} Q^1_{MC,k(1)} \\ \cdots \\ Q^r_{MC,k(r)} \end{pmatrix}.$$

Let B_{full} be a vector of statistics for feature selection based on the full dataset. Then AROHIL modeling can use the hierarchical model,

$$B_{\mathrm{full}} \sim F_B$$
$$\begin{bmatrix} \tilde{Q}^1_{\mathrm{AROHIL}} \\ \cdots \\ \tilde{Q}^r_{\mathrm{AROHIL}} \end{bmatrix} \Bigg| B_{\mathrm{full}} \sim F_Q(\cdot|B_{\mathrm{full}}).$$

In the working example, B_{full} is the vector of pooled variance t-statistics. The gene selection model reduces the dimension of the space using B_{full}. The elements of Q^i_{AROHIL} are then $\hat{L}_k$ elements, plus a classification cutpoint. The $\hat{L}_k$ elements are generated from truncated Student's t distributions with $n-3$ degrees of freedom and non-centrality parameter calculated from the model. (Another approach would be to generate the $\hat{L}_k$ elements by adding noise to the corresponding elements of B_{full}. But this led to computational problems due to the fact that the sum must be truncated so as to ensure coherence with the selection model.) This leads to the approach that Q^i_{AROHIL} is generated by:

$$\mathrm{ncp} = \frac{\left(\mu_x - \mu_y\right)' D^{-1}}{\sqrt{4/n}}$$
$$Q^i_{\mathrm{AROHIL}} \sim \mathrm{trunc}MVT\left(t_{\alpha/2,n-3}, \mathrm{ncp}, n-3\right)$$

where $\mathrm{trunc}MVT(a,b,c)$ is a vector of independent Student T random variables, truncated away from zero at $\pm a$, with non-centrality vector b, and with degrees of freedom c. See Section 5 in Supplementary Material for further details.

3 RESULTS

AROHIL Monte Carlo was applied in multiple places to the algorithm of Dobbin (2009). Briefly, the method of Dobbin (2009) constructs a lower confidence bound for the true prediction accuracy of a classifier developed on high-dimensional data. This bound provides an estimate of the variability in the leave-one-out estimate of prediction accuracy which is otherwise problematic to assess. The motivation was to convert the method of Dobbin (2009) from a set of C++ programs, to a single R program, while at the same time reducing the computation time. We first compare pure Monte Carlo to AROHIL Monte Carlo computation times in some simple settings. Then we turn to the implementation of the method of Dobbin (2009) and analyze some of the results to validate the AROHIL method, and to show how intermediate steps of AROHIL Monte Carlo algorithms can be checked.

We first performed a set of simulations to benchmark the computational savings of AROHIL Monte Carlo compared with traditional Monte Carlo on some simple examples. Results are presented in Table 2. As can be seen from the table, computational costs in high dimensions can be reduced several orders of magnitude by using AROHIL Monte Carlo instead of traditional Monte Carlo. On the fourth row of the table, representing a 5000 dimensional space, the computation time is reduced from over a day to under a minute. Also note that the estimates from the two methods are practically identical.

Now we turn to the AROHIL program implementation. Three key intermediate steps in the application of the AROHIL method to the algorithm of Dobbin (2009) are as follows:

(1) Generate the distribution of cutpoints used for classification of samples.

(2) Generate mean accuracies corresponding to a particular high-dimensional Mahalanobis distance between the class means.

Table 2. Benchmark simulations

Sim	P	Computing time			Estimate	
		Pure MC	AROHIL MC (s)	Times ratio	Pure MC	AROHIL MC
No Resampling	5000	32 min 45.13 s	0.50	3930	0.753 (0.032)	0.753 (0.034)
No Resampling	1000	4 min 55.9 s	1.11	267	0.816 (0.020)	0.818 (0.022)
No Resampling	100	25.65 s	0.70	37	0.837 (0.010)	0.838 (0.009)
Resampling	5000	1 day 12 h 5 min 28.01 s	53.01	2451	0.850	0.850
Resampling	1000	7 h 11 min 12.7 s	33.94	762	0.883	0.883
Resampling	100	43 min 27.07 s	28.97	90	0.900	0.900

Benchmark comparison of computation times. Computing times are total elapsed times in seconds. Times ratio is $\frac{Pure\ MC}{AROHIL\ MC}$. When Sim = No Resampling, simulations are estimating the mean classification accuracy for fixed values of the parameters, with $\alpha=0.001$, $\delta=1$, P dimensions, $n=60$, using CCP predictor over 500 simulations. When Sim = Resampling, simulations are estimating the 90th percentile of the leave-one-out cross-validated prediction accuracy distribution for fixed values of the parameters, with $\alpha=0.001$, $\delta=1$, P dimensions, $n=60$, using CCP prediction over 1000 simulations. No Resamplings done in R on a 32-bit operating system, Resamplings done in R on a 64 bit operating system. No Resampling estimates include standard deviations in parentheses.

(3) Generate the lengths of predictors, $\left|\hat{L}\right|$ conditional on a true accuracy a_{true} of the full-dataset predictor.

Each of these steps was reimplemented using AROHIL Monte Carlo and these intermediate results were carefully checked. Results are presented in Section 3 in Supplementary Material.

The critical test is the performance of the AROHIL Monte Carlo method in terms of maintaining coverage probabilities compared with the method of Dobbin (2009). This requires estimating quantiles of the leave-one-out cross-validation distribution. As shown in Table 3, coverage probabilities are quite close to nominal over a wide range of settings, and appear to improve over Dobbin (2009) by being less conservative.

The AROHIL method was used to evaluate coverage probabilities in the presence of violations of model assumptions. Results are shown in Table 4. As can be seen in the table, the coverage probabilities do break down in extreme cases, particularly for very heavy tailed distributions, such as the Student's t distribution with 1 degree of freedom (where the variance is infinite). But overall the method is quite robust.

In Table 5, the method was applied to five datasets evaluated in Michiels *et al.* (2005) to construct lower confidence bounds for prediction accuracy. Datasets were downloaded from the BRB Array Tools data archive (Zhao and Simon, 2008). Note that for three of the five datasets, a 90% lower confidence bound does not contain 50%, indicating better than chance classification. The more conservative 97.5% lower bound is above 50% for two of the datasets. This is in contrast to Michiels *et al.* (2005), in which all their 95% two-sided intervals (equivalent to our one-sided 97.5% interval) contained 50%. Importantly, these two datasets (Rosenwald *et al.*, 2002 and van't Veer *et al.*, 2002) have found supporting evidence in subsequent publications (Bea *et al.*, 2005 and Wittner *et al.*, 2008), which seems to suggest that the method of Michiels *et al.* (2005) is overly conservative compared with our AROHIL Monte Carlo.

4 DISCUSSION

We have presented a mathematical modeling approach to speed up high-dimensional Monte Carlo simulations by reducing the effective dimension of the space in which the simulations are performed. We have described in a general way how this approach can be used in the case of simple Monte Carlo simulations, and also Monte Carlo simulations that require resampling, such as bootstrap or cross-validation. The modification for the resampling setting is achieved by constructing a hierarchical model for which the distributions of the functionals of interest match (or approximately match) the pure Monte Carlo distributions. This new method is called AROHIL, and can enable complex and slow high-dimensional simulations to be converted into simpler and much faster low-dimensional simulations. We have discussed how this method can be used to improve robustness evaluations and to disseminate software. As an example, we are disseminating an AROHIL program with this article, and have presented a robustness evaluation of this previously published method. In the discussion below, we discuss AROHIL Monte Carlo generally first, and then the implementation program provided in this article.

We have discussed one detailed example of how high-dimensional leave-one-out cross-validation Monte Carlo can be converted into an AROHIL Monte Carlo. Generalizing this to other cross-validations, such as 10-fold cross-validation, is straightforward. Bootstrapping by AROHIL would require a further modification. We showed in this article that AROHIL for cross-validation is performed by calculating the distribution of a backbone vector of statistics

Table 3. Coverage probability assessment

DE features	MAHD/2	$\bar{a}$	90% LB coverage	90% LB naive bin coverage
1	2.5	99%	1.000	1.000
1	2.0	95%	0.895	0.860
1	1.5	87%	0.918	0.917
1	1.0	73%	0.902	0.864
5	2.5	99%	1.000	1.000
5	2.0	95%	0.941	0.962
5	1.5	87%	0.931	0.925
5	1.0	70%	0.897	0.867

Coverage probabilities calculated from 1000 replications. Model settings are $P=1000$ features, $n_1=n_2=30$ per class,'DE Features' are differentially expressed features; compound covariate predictor with gene selected based on significance cutoff $\alpha=0.011$.

Table 4. Robustness assessment of coverage probabilities

Distn	Σ	DEG	δ	ρ	$\bar{a}$	90% LB coverage	
						AROHIL	Bin
MVN	CS	5	3.0	0.6	0.94	0.97	0.97
MVN	CS	5	2.0	0.6	0.84	0.98	0.95
MVN	CS	5	1.0	0.6	0.65	0.90	0.88
MVN	AR(1)	30	4.0	0.8	0.92	1.00	0.97
MVN	AR(1)	30	3.0	0.8	0.85	0.99	0.98
MVN	AR(1)	30	2.0	0.8	0.72	0.96	0.95
MVN	Emp	1	3.0	NA	0.90	0.97	0.96
MVN	Emp	1	2.0	NA	0.79	0.95	0.92
MVN	Emp	1	1.0	NA	0.60	0.87	0.85
MVT1	Diag.	1	5.0	0	82%	0.934	0.940
MVT1	Diag.	1	3.0	0	65%	0.876	0.869
MVT1	Diag.	1	2.0	0	58%	0.866	0.870
MVT3	Diag.	1	3.0	0	95%	0.960	0.989
MVT3	Diag.	1	2.0	0	85%	0.928	0.912
MVT3	Diag.	1	1.0	0	63%	0.878	0.851

Coverage probabilities under model violations. Columns are as follows: ‘Distn’ is the high-dimensional distribution of the data, multivariate normal or multivariate T with 1 or 3 degrees of freedom. Σ is the form of the covariance matrix, CS for block compound symmetric, AR(1) for block autoregressive order 1, Empirical for empirically estimated using shrinkage from Rosenwald *et al.* (2003) data, Diag for diagonal. ‘DEG’ is the number of differentially expressed features. $\delta/\sqrt{DEG}$ is the effect size for individual differentially expressed features. ρ is the correlation parameter for CS and AR(1). ‘$\bar{a}$" is the mean true accuracy over all simulations. ‘90% LB Coverage’ is the coverage probability of 90% lower confidence bound, using either AROHIL or an exact binomial confidence interval constructed (naively/incorrectly) from the LOOCV accuracy estimate.

Table 5. Applications to real datasets

Dataset	Dim	Dead	Alive	$\hat{a}_{\text{loocv}}$	90% LB	97.5% LB
Rosenwald	7399	112	116	61%	0.53	0.53
van’t Veer	24 481	51	46	65%	0.58	0.52
Beer	7129	24	60	56%	<0.50	<0.50
Pomeroy	7129	21	39	62%	0.52	<0.50
Bhattacharjee	12 600	53	74	53%	<0.50	<0.50

Applications to real datasets used in Michiels *et al.* (2005). $\hat{a}_{loocv}$ is the leave-one-out cross-validation accuracy. *Dim* is the number of features, n_1 and n_2 are the number from each class. ‘90% LB’ is the 90% lower confidence bound computed by AROHIL Monte Carlo; and similarly ‘97.5% LB’ is a 97.5% LB, comparable to the 95% two-sided intervals used in Michiels *et al.* (2005) . For the Rosenwald *et al.* (2002) dataset, the outcome is survival status at 3 years. For the van’t Veer *et al.* (2002) dataset, outcome is 5-year metastases-free survival. For the Beer *et al.* (2002) dataset, outcome is survival status. For the Pomeroy *et al.* (2002) dataset, outcome is survival status. For the Bhattacharjee *et al.* (2001) dataset, outcome is survival status. For all datasets, the significance level for gene selection was $\alpha = 0.001$.

that represents the full dataset, then calculating the conditional distribution of key cross-validation statistics when a sample is left out. For bootstrapping, an extra level would need to be added to the hierarchical model that would represent the overlap pattern between the bootstrap samples. This pattern could be represented by a simple multinomial model with probability $1/n$ on each of the n samples for each of the bootstrap draws (sampling with replacement). Then the conditional distribution given the backbone vector and the pattern can be derived in a straightforward way and used to generate the bootstrap sample.

We have discussed that sometimes AROHIL models will require approximations to the pure Monte Carlo distribution. Importantly, such approximations must be checked carefully to ensure that they are true to the original model. On the other hand, it does not seem reasonable to ‘throw the baby out with the bathwater’ and abandon AROHIL Monte Carlo when any approximations are required. In many cases, these approximations are straightforward to check over the range of simulation settings that are of interest.

We have termed the dimension reduction step of AROHIL as adequate, and not attempted here to define this idea exactly. Dimension reduction could be based on more general notions such as sufficiency. A potential area of future research is to find a more formal approach to the dimension reduction step which would establish that the statistics used by the AROHIL Monte Carlo are capturing all the key aspects of the pure Monte Carlo.

A potential critique of the AROHIL approach is that it requires some work to build the mathematical models used to reduce dimension. While it is true that this method requires some extra work, which is not generally worth the trouble in lower dimensional settings, the computational savings in high dimensions is so large that it can not only be worthwhile but also critical. Furthermore, very complex high-dimensional procedures can be challenging to implement, and thoroughly checking for coding bugs, information leak or inadvertent neglect of specification of all parameters and assumptions, can be fraught with difficulties. An important aspect of AROHIL is that implementation is simplified, i.e. the added complexity of the mathematics is often more than compensated for by the greater simplicity and transparency of the computer code. We argue that this results in a cleaner and overall simpler procedure than traditional brute force Monte Carlo, where any errors are often buried in long computer code scripts.

We have found that the AROHIL Monte Carlo approach results in very short and simple code compared with pure Monte Carlo. For example, the R script we are providing with this publication is much shorter and simpler than the original code from Dobbin (2009), consisting of multiple C++ programs and steps to integrate the outputs together. The resulting simplification of the code is likely to greatly enhance reproducibility of high-dimensional studies, which has been a continuing challenge to this area.

The accompanying AROHIL program is implemented with one informative feature, which was used in all the coverage probability simulations in this article. The program is also available with a user-selected number of features. The number of informative features has relatively small effect on the confidence interval bound, and the number of informative features is unknown. Hence, it is preferable to have a program in which the user does not have to come up with this unknown quantity. See Section 6 in Supplementary Material for the table of simulation results showing the stability of bounds across different numbers of informative features. An alternative approach would be to search over different possible numbers of informative features to find a worst-case scenario setting, resulting in more conservative confidence bounds.

The accompanying AROHIL program is implemented with a diagonal covariance matrix. This is done not because the true covariance for high-dimensional data is likely to be diagonal, but because it is generally not possible to estimate the covariance matrix well in high dimensions. Covariance matrices may be estimated with a shrinkage estimate of the form $\hat{\Sigma}_{shrink} = w \times Diag(\hat{\Sigma}) + (1-w) \times \hat{\Sigma}$ (Ledoit and Wolfe, 2004; Schafer and Strimmer, 2005), as was done for Table 3. An area of potential future work is to use covariance estimates to better tune this method to a given set of data.

Finally, there are a few additional comments on the program implementation provided in this article. This program provides a lower confidence bound on prediction accuracy, taking as input the observed leave-one-out cross-validated accuracy, the dimension of the feature space, the number of differentially expressed features and the stringency used for feature selection. This program implements the method of Dobbin (2009), which is a Monte Carlo-based method that assumes a multivariate normal distribution. While the simulations presented in this article and Supplementary Material suggest that the method is quite robust to model violations, it should be noted that the gold standard in high dimensions is generally considered to be non-parametric resampling-based methods. For example, Jiang *et al.* (2008) have developed a bootstrap method for constructing confidence intervals for prediction accuracy. This bootstrap method could serve as a more robust check on the interval constructed with the AROHIL program provided here. One can also note that in evaluating whether a classifier is statistically significantly better than chance, permutation tests (Westfall and Young, 1993) are probably more appropriate than confidence-bound-based approaches. One should also not use this method in a vacuum, but note that previous work has been done. For example, Mukherjee *et al.* (2003) and Dobbin *et al.* (2008) have presented sample size guidelines for studies that may indicate whether one should expect a confidence bound to be reasonably tight. Also, if the class prevalences are highly imbalanced (e.g. 90% from one class and 10% from the other), then overall classification accuracy is probably not as important as other quantities, such as positive or negative predictive values. See Dobbin and Simon (2011) for discussion.

ACKNOWLEDGEMENTS

We thank the referees who provided insightful suggestions that improved this article.

Funding: Georgia Cancer Coalition through the Distinguished Cancer Clinicians and Scientists Program (to K.K.D. and S.C.).

Conflict of Interest: none declared.

REFERENCES

Bea,S. *et al.* (2005) Diffuse large B cell lymphoma subtypes have distinct genetic profiles that influence tumor biology and improve gene expression-based survival prediction. *Blood*, **106**, 3183–3190.

Beer,D.G. *et al.* (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, **8**, 816–824.

Bhattacharjee,A. *et al.* (2001) Classification of human lung adenocarcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl Acad. Sci. USA*, **98**, 13790–13795.

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

Dobbin,K.K. and Simon,R.M. (2011) Optimally splitting cases for training and testing high dimensional classifiers. *BMC Med. Genom.*, **4**, 31.

Dobbin,K.K. *et al.* (2008) How large a training set is needed to develop a classifier for microarray data? *Clin. Cancer Res.*, **14**, 108–114.

Dudoit,S. *et al.* (2003) Multiple hypothesis testing in microarray experiments. *Stat. Sci.*, **18**, 71–103.

Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **6**, 721–774.

Jiang,W. *et al.* (2008) Calculating confidence intervals for prediction error in microarray classification using resampling. *Stat. Appl. Genet. Mol. Biol.*, **7**, Article 8.

Ledoit,O. and Wolf,M. (2004) A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivar. Anal.*, **11**, 365–411.

Michiels,S. *et al.* (2005) Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet*, **365**, 488–492.

Mukherjee,S. *et al.* (2003) Estimating dataset size requirements for classifying DNA microarray data. *J. Comput. Biol.*, **10**, 119–142.

Nguyen,D.V. and Rocke,D.M. (2004) On partial least squares dimension reduction for microarray-based classification: a simulation study. *Comp. Stat. Data Anal.*, **3**, 407–425.

Olshen,A.B. *et al.* (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.

Pomeroy,S.L. *et al.* (2002) Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature*, **415**, 436–442.

Radmacher,M.D. *et al.* (2002) A paradigm for class prediction using gene expression studies. *J. Comput. Biol.*, **9**, 1462–1469.

Ripley,B.D. (1987) *Stochastic Simulation*. Wiley and Sons, New York.

Robert,C.P. and Casella,G. (2004) *Monte Carlo Statistical Methods*, 2nd edn. Springer, New York.

Rosenwald,A. *et al.* (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med.*, **346**, 1937–1947.

Schafer,J. and Strimmer,K. (2005) A shrinkage approach to large-scale covariance matrix estimation and applications for functional genomics. *Stat. Apps. Genetics Mol. Biol.*, **4**, Article 32.

van't Veer,L.J. *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Venkatraman,E.S. and Olshen,A.B. (2007) A faster circular binary segmentaion algorithm for the analysis of array CGH data. *Bioinformatics*, **23**, 657–663.

Westfall,P.H. and Young,S.S. (1993) *Resampling-Based Multiple Testing*. Wiley, New York.

Wittner,B.S. *et al.* (2008) Analysis of the MammaPrint breast cancer assay in a predominantly postmenopausal cohort. *Clin. Cancer Res.*, **14**, 2988–2993.

Zhao,Y. and Simon,R. (2008) BRB ArrayTools Data Archive for Human Cancer Gene Expression: A Unique and Efficient Data Sharing Resource. *Cancer Informatics*, **6**, 9–15.

Zou,H. and Hastie,T. (2005) Regularization and variable selection via the elastic net. *J. R. Stat. Soc. B*, **67**, 301–320.