Gene Selection Using Iterative Feature Elimination Random Forests for Survival Outcomes

Herbert Pang, Stephen L. George, Ken Hui, and Tiejun Tong

Abstract—Although many feature selection methods for classification have been developed, there is a need to identify genes in high-dimensional data with censored survival outcomes. Traditional methods for gene selection in classification problems have several drawbacks. First, the majority of the gene selection approaches for classification are single-gene based. Second, many of the gene selection procedures are not embedded within the algorithm itself. The technique of random forests has been found to perform well in high-dimensional data settings with survival outcomes. It also has an embedded feature to identify variables of importance. Therefore, it is an ideal candidate for gene selection in high-dimensional data with survival outcomes. In this paper, we develop a novel method based on the random forests to identify a set of prognostic genes. We compare our method with several machine learning methods and various node split criteria using several real data sets. Our method performed well in both simulations and real data analysis. Additionally, we have shown the advantages of our approach over single-gene-based approaches. Our method incorporates multivariate correlations in microarray data for survival outcomes. The described method allows us to better utilize the information available from microarray data with survival outcomes.

Index Terms—Cancer, gene selection, iterative feature elimination, microarrays, random forest, survival.

1 Introduction

The identification of genes with high prognostic value and predictive power in clinical trials may help patients with cancer and other diseases. This may result from discovery of genes that serve as novel drug targets or those that can help guide better therapeutic decisions.

Many gene selection methods in the classification setting have been proposed in recent years. One of the earlier approaches was developed by Díaz-Uriarte and Alvarez de Andrés [1] for selecting genes with a random forests recursive feature elimination algorithm for class prediction. Duan et al. [2] proposed a recursive support vector machine gene selection algorithm. Tang et al. [3] described a two-stage recursive feature elimination strategy for gene selection. Niijima and Okuno [4] extended the Laplacian linear discriminant analysis and developed a new algorithm for unsupervised feature selection. Mao and Tang [5] proposed a regularized recursive Mahalanobis separability measure for gene selection for classification problems. Mundra and Rajapakse [6] developed a filter-based support vector machine recursive feature elimination based on mutual information for the same purpose. Luo et al. [7] proposed an improved support vector machine recursive cluster elimination that utilizes clusters to further improve gene selection performance for the classification setting.

Bøvelstad et al. [8] have pointed out the importance of survival prediction. There are some implicit feature selection techniques such as boosting for survival outcomes [9]. Others developed Bayesian approach using model averaging [10]. However, few methods of multivariate gene selection have been developed for censored survival data. Current approaches are mostly univariate approaches, and the number of genes is usually chosen with an arbitrary cutoff. The purpose of this paper is to introduce an iterative feature elimination algorithm for gene selection using random survival forests. The technique of random forests is among the best for correlating survival time with microarray data [11], [12]. We extend existing methods to survival phenotype and compare our method with other machine learning approaches that are known to do well in the survival setting. Our approach provides an alternative to univariate gene selection (such as Dunkler et al. [13]) for survival outcomes, with enhanced operating characteristics allowing the researchers to focus on a smaller set of genes for high-dimensional data.

- *H. Pang is with the Department of Biostatistics and Bioinformatics, Duke University School of Medicine, DUMC 2721, Hock Plaza 11099, Durham, NC 27705. E-mail: herbert.pang@duke.edu.*
- *S.L. George is with the Department of Biostatistics and Bioinformatics, Duke University School of Medicine, Room 8037, Suite 802, 2424 Erwin Road, Durham, NC 27705-3833. E-mail: Stephen.george@duke.edu.*
- *K. Hui is with the Department of Internal Medicine, Yale University School of Medicine, 1 Gilbert Street, TAC S160, New Haven, CT 06520. E-mail: ken.hui@yale.edu.*
- *T. Tong is with the Department of Mathematics, Hong Kong Baptist University, FSC 1202, Kowloon Tong, Hong Kong. E-mail: tongt@hkbu.edu.hk.*

Manuscript received 15 Sept. 2010; revised 20 Feb. 2012; accepted 4 Apr. 2012; published online 26 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2010-09-0221. Digital Object Identifier no. 10.1109/TCBB.2012.63.

2 Methods

2.1 Random Forests

The random forest technique was first proposed for classification and regression settings [14]. The first version of survival forests, random forests for censored data, was implemented in [15]. The key difference between random

forests classification/regression and its survival counterpart is that for random survival forests [16], the outcome of interest is a set of survival times with a corresponding censoring indicator. As a result, the split criteria for censored survival data are different from standard random forests for classification and regression, which uses the Gini criterion [14].

These differences are outlined in more detail in Section 2.2. However, the main properties of the original random forests are preserved in the random survival forests algorithm. For gene expression data, we treat each individual gene as a continuous intensity value. A random survival forest encompasses many binary trees, each formed by a deterministic algorithm. First, each tree is built using a bootstrap sample of the patients of interest. Second, a best binary split is chosen using a subset of genes in the data set. Unlike classification and regression trees (CARTs), no pruning is involved. Several split criteria are available in random survival forests; we apply log-rank (LR), log-rank score (LRS), conserve, and random, for split criteria as described in Section 2.3. Each observation is assigned to a leaf, the terminal node of a tree, according to the order and values of the predictor variables.

2.2 Random Survival Forests Algorithm

1. Draw bootstrap samples from the original data *ntree* times, where ntree is the number of trees. For each bootstrap sample, some observations are left out-of-bag (OOB).
2. A binary survival tree is grown for each bootstrap sample.
3. Let p be the number of genes in the data set; at each node of the tree, the number of genes selected at random for splitting will be $p^{1/2}$. This choice as suggested by Lunetta et al. does not have strong influence on the prediction error and variable importance [14], [15], [17].
4. Using one of the split criteria described in Section 2.3, a node is split using the single gene from the $p^{1/2}$ randomly chosen genes that maximizes the survival difference between the children nodes.
5. The splitting continues until each terminal node size reaches the minimum number of events with unique survival times. The default is three for right censored data [16].
6. The binary survival trees are then aggregated to obtain the ensemble cumulative hazard estimates as detailed below.

The binary trees are aggregated to form the forest through using ensemble cumulative hazard function (eCHF). The idea is essentially grouping the hazard estimates from the terminal nodes. The CHF estimates for a terminal node L is the Nelson-Aalen estimator

$$\hat{\Lambda}_L(t) = \sum_{t_{i,L} \le t} \frac{d_{t_i,L}}{R_{t_i,L}}, \tag{1}$$

where $t_{i,L} =$ distinct survival time, $d_{t_i,L} =$ the number of events and $R_{t_i,L} =$ the number of individuals at risk at time $t_i,_L$. For every binary survival tree with Q terminal nodes, there will be Q different CHF estimators. The CHF estimate for an individual $\mathrm{i_{new}}$ with gene predictor $\mathrm{gene_{new}}$ can be found by identifying which terminal node includes the individual. That is, the CHF estimate is equal to $\hat{\Lambda}_L(t)$ if $\mathrm{i_{new}}$ ended in terminal node L. The eCHF is simply the sum of the CHFs across the bootstrap samples divided by number of trees. Let n be the total number of individuals, the expected number of ensemble events can be obtained by summing over time T_j for $j = 1$ to n.

The OOB error rate is based on a prediction measure called Harrell's concordance index, denoted as C [18]. The OOB error rate, defined as $1 - \mathrm{C}$, where $\mathrm{C} \in$ [0,1], measures how well the random forests correctly ranks the survival of any pair of individuals. A value of $(1 - \mathrm{C})$ of 0.5 corresponds to a random guess and 0 means perfect concordance. The random survival forests implements both "randomsplit" and "permute" for variable importance. The latter is the same way Breiman used to perturb a variable in the original Random Forests algorithm [14]. The variable importance for each gene, used in Step 1) of Section 2.4, is calculated by permutation. The random survival forests algorithm permutes the values of the gene in the OOB cases, and the cases with permuted values are dropped down their in-bag survival tree. The cumulative hazard function is then calculated for each tree and aggregated across the trees. The randomly permuted values of the gene in the OOB individuals and the outcome of interest are independent of each other. The variable importance for a predictor x is equal to PEo-PEn, where PEo is the prediction error of the original ensemble and PEn is the prediction error of the new ensemble. Variable importance is based on the concordance index described above and measures the extent of misclassification when the variable of interest becomes independent of outcome for the random survival forests.

2.3 Split Criteria

We will describe two split criteria that performed well in our study. Let $i(i = 1, \ldots, n)$ denotes a single individual; and x denotes one of the genes. The proposed split is of form $\mathrm{x} \le \mathrm{c}$ and $\mathrm{x} > \mathrm{c}$, where c is the cutoff value.

Log-rank. The log-rank split criterion (2) [19], which measures the node separation, is based on the log-rank test statistic defined as

$$\mathrm{LR(X,c)} = \frac{\sum_{i=1}^{E} d_{t_i,child_1} - E(D_i)}{\sqrt{\left[\sum_{i=1}^{E} \mathrm{var}(D_i)\right]}}, \tag{2}$$

where E is the number of distinct event times in the parent node, $d_{t_i,child_j}$ is the number of events at time $\mathrm{t_i}$ in the child nodes $j = 1, 2$, and $R_{t_i,child_j}$ is the number of individuals at risk at time $\mathrm{t_i}$ in the child nodes $j = 1, 2$, and $R_{t_i} = \sum_{j=1}^{2} R_{t_i,child_j}$, $d_{t_i} = \sum_{j=1}^{2} d_{t_i,child_j}$, D_i, is the random variable corresponding to the number of events in child node $j = 1$ for the ith distinct event time, $E(D_i) = R_{t_i,child_1}(d_{t_i}/R_{t_i})$ is the expectation, $Var(D_i) = d_{t_i}(R_{t_i} - d_{t_i})/(R_{t_i} - 1) \times R_{t_i,child_1}/R_{t_i} \times (1 - R_{t_i,child_1}/R_{t_i})$ is the variance. The best split is defined as the one that maximizes the absolute value of the equation above.

Log-rank score. Let $n_1 = \sum_{i=1}^{n} \mathbf{I}(X_i \le c)$, where $\mathbf{I}(.)$ is an indicator function. The log-rank score split criterion (3), which measures the node separation, uses the following equation:

$$\mathrm{LRS(X,c)} = \frac{\sum_{X_i \le c} a_i - n_1 \bar{a}}{\sqrt{n_1\left(1 - \frac{n_1}{n}\right) s_a^2}}, \quad (3)$$

where for individual i, $a_i = \mathbf{I}_i - \sum_{l=1}^{\gamma_i} \frac{\mathbf{I}_l}{n-\gamma_l+1}$, as defined in [22], where $\mathbf{I}_i = 1$ if an event is observed for individual i and 0 otherwise; $\gamma_l = \sum_i^n \mathbf{I}(S_i \le S_l)$ is the number of observed events or censored, occurring at survival time S_l or before; and $\bar{a}$ and ${s_a}^2$ are the sample mean and sample variance of a_i, respectively. The best split is defined as the one that maximizes the absolute value of the LRS equation above.

Conserve split criterion. Another type of splitting rule is the conservation of events [23]. Denote the Nelson-Aalen cumulative hazard estimator (4) for child j as

$$\hat{\wedge}_j(t) = \sum_{t_1, j \le t} \frac{d_{t_1, child_j}}{R_{t_1, child_j}}, \quad (4)$$

where $t_{i,j}$ are the ordered event times for child j.

The conservation of events asserts that the total number of events is conserved in each child (5), i.e.,

$$\sum_{i=1}^{n_j} \Lambda(T_{i,\, child_j}) = \sum_{i=1}^{n_j} 1_{i, child_j}(censoring), \quad (5)$$

where $1_{i,child_j}(censoring)$ is the censoring indicator.

Let $T_{(1),child_j} \le T_{(2),child_j} \le \cdots \le T_{(n),child_j}$ be the ordered time points for child j, and $1_{(i),child_j}(censoring)$ be the corresponding censoring indicator for $T_{(i),child_j}$, for $k = 1, \ldots, n_j$

$$Con(X,c) = \frac{1}{R_{t_i, child_1} + R_{t_1, child_2}} \sum_{j=1}^{2} R_{t_1, child_j} \sum_{k=1}^{n_j - 1} |M_{k,j}|, \quad (6)$$

where

$$M_{k,\, child_j} = \sum_{i=1}^{k} \hat{\Lambda}\left(T_{(i),\, child_j}\right) - \sum_{i=1}^{k} \mathbf{1}_{(i),\, child_j}(censoring).$$

The Con(X,c) (6) measures whether the two groups are well separated and $(1 + \mathrm{Con(X,c)})^{-1}$ is used in the program. It finds the best split by finding children closest to the conservation of events principle.

Random split criterion. The last splitting rule is "random" which implements a purely random uniform splitting [24]. For each node, a variable is randomly selected from a random set of $\sqrt{p}$ variables. For a chosen split variable X, a random split point is chosen among all possible split points on that variable.

2.4 Gene Selection with Random Survival Forests

The schema for the random survival forests gene selection algorithm is presented in Section A of the supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.63. A description is as follows:

1. Gene expression and survival data are fed as input to the random survival forests algorithm. For each gene, $\mathrm{v_1, v_2, v_3, \ldots, v_p}$, the variable importance is calculated.
2. For iteration i, order the genes by variable importance in descending order and remove the bottom 20 percent (default). 20 percent is also the default chosen by [1] Díaz-Uriarte and Alvarez de Andrés and real data analysis also suggests that this is a good choice, see Section 5.1. Call this removed set $\mathrm{gs_i}$.
3. Calculate the OOB error rate (1-C as defined above) associated with the set $\mathrm{gs_i}$ using the expected number of ensemble events from random survival forests and the observed survival outcome.
4. Repeat Step 2 until $\mathrm{gs_i}$ contains only two genes. This is the limit for inclusion if we consider a multivariate gene selection.
5. Find the set of genes with the minimum number of genes such that the OOB error rate is within 1 standard error (s.e.), default, of the minimum OOB error rate, mOOB, of all iterations. That is $\mathrm{argmin_i}\ \{\#\mathrm{gs_i}\} : \mathrm{mOOB} - \mathrm{s.e.} \le \mathrm{OOB_i} \le \mathrm{mOOB} + \mathrm{s.e.}$ This one s.e. rule is commonly used in the classification tree literature [1], [23], [24]. The sampling error is determined by $\mathrm{OOB_i}$ and the total number of subjects.

2.5 Estimation of Error Rates

Given the iterative nature of the approach, the OOB error underestimates the true error rate and cannot be used to assess the overall error rate. This is analogous to the reasons that lead to selection bias [25], [26]. Therefore, we will use the 10 times 10-fold cross-validation, in which the gene selection procedure is applied to each training sample. The selected genes will then be used to predict the expected number of ensemble events in the testing set to assess prediction error rates. However, from the perspective of selecting the final optimal number of genes, incorporating OOB error rates to perform gene selection is not necessarily a bad procedure [1], [27].

3 Other Machine Learning Methods

3.1 Random Conditional Inference Forests

Conditional inference forests [28] (R package party [29]) involves building many conditional inference trees under a binary recursive partitioning algorithm. The first step involves variable selection by testing a global null hypothesis of independence between any of the predictors and the survival outcome of interest. The algorithm terminates if the hypothesis cannot be rejected, otherwise, the predictor with the strongest association to the survival outcome will be selected. A binary split on the log-rank score split criterion is implemented using the selected predictor. The steps above are then recursively repeated.

3.2 Survival Neural Network

The neural network for survival analysis [30] (R package survnnet [31]) utilizes the commonly used structure of a feed-forward network. A neural network consists of an input layer, hidden layers, and an output layer. Each layer is composed of several neurons which receive output from neurons in the previous layer. The output is generated based on a specific transformation. Weights are applied to the input values between the layers. The goal of a neural network procedure is to seek values for the weights to fit the training data well. In the training procedure, the weights are iteratively updated to minimize the differences

of the observed output and the generated output. This is usually done using a back propagation algorithm. We use parametric survival net, fitting a Weibull distribution to the data. The log hazard is modeled as a function of covariates, input, with a single linear output. Other settings include three hidden layers, Weibull shape parameter $\mathrm{p} = \exp(0.1)$, and skip-layer units set to true.

3.3 Survival Cox Boosting

Given a set of outcome and predictors, the goal is to approximate the outcome with a function. Cox boosting (R package mboost [32]) proceeds as follows to optimize a loss function. First, initialize the function to a constant offset. Second, compute the residuals defined as the negative partial log-likelihood for Cox models. Third, fit the negative gradient vector by a base procedure, in this case, a component-wise univariate linear model as described in [33]. Fourth, update the function by taking a step of size 0 to 1 of the base learner. The second to fourth steps are repeated until a predetermined number of iterations are completed. The boosting algorithm is based on functional gradient descent. For more details see [34], [35].

3.4 Survival Support Vector Machine

The goal of the support vector machine for survival (R package survpack [36]) is to find a nonlinear hyperplane that separates the individuals who had events before time *t* and those without an event at that point. The nonlinear relationship is obtained from a reproducing kernel Hilbert space with a Gaussian kernel of width 1. This procedure is repeated at every event time. As in soft-margin support vector machines, the survival counterpart allows for cases when the hyperplanes do not exist by penalizing observations that lie on the other side of the margin. The minimization problem involves a sum that is based on the concordance index [37]. The penalty term is set to 1. The optimization problem is a positive-definite quadratic problem and is solved by using sequential minimal optimization algorithm.

3.5 Gene Selection with Random Conditional Inference Forests or Survival Neural Network

To compare with our approach described Section 2.4, Random Conditional Inference Forests is incorporated with Cox regression gene selection following a similar scheme for the iterative feature elimination algorithm. Similarly, this is done for Survival Neural Network. A description is as follows:

1. For each gene, univariate Cox regression is run.
2. For iteration *i*, order the genes by p-values in ascending order and remove the bottom 20 percent.
3. Calculate the concordance index associated with the set $\mathrm{gs_i}$ using the predicted survival time from Random Conditional Inference Forests or Survival Neural Network.
4. Repeat Step 2 until $\mathrm{gs_i}$ contains only two genes.
5. Find the set of genes with the minimum number of genes such that the error rate is within 2 s.e. Two is used because error rate tends to be the lowest with the largest number of genes chosen based on p-values from univariate Cox regression with either conditional inference forest or survival neural network.

TABLE 1
Data Sets

Data set	Reference	Sample Size	Genes in Cancer Pathways	All Genes	Response
MCL	Rosenwald et al. (2003)	92	5969	8810	Overall Survival
DCL	Rosenwald et al. (2002)	240	1320	7399	Overall Survival
NKI	van de Vijver et al. (2002)	295	2740	9746	Overall Survival

4 Data Sets

4.1 Real Data Sets

The data sets used are given in Table 1. To improve the efficiency of the procedure, genes were filtered to contain only those that are mapped to known genes in cancer pathways [http://www.broad.mit.edu].

4.2 Simulation Data Sets

Our simulation design is to investigate the performance of our algorithm both under the null of no signal and under the alternative with different settings. This setup is similar to [1]. We simulated data under various settings with different numbers of independent dimensions, $\mathrm{d} = 1, 2, 3,$ and 10, $\mathrm{b} = 100$, and different number of genes per dimension, $\mathrm{g} = 5, 20, 100$. The sample size n is 50. The genes in each dimension are simulated from a multivariate normal distribution $\mathrm{MVN}(0, \Sigma)$.

All genes have a variance of 1 from the multivariate normal distribution, and the correlation between genes within a dimension is 0.9, whereas the correlation between genes among dimensions is 0. In other words, the covariance matrix Σ is a block-diagonal matrix of size $\mathrm{d} * \mathrm{b}$ by $\mathrm{d} * \mathrm{b}$ with each of the b by b diagonal blocks Σ_ρ, and the rest of the matrix is zero. After the genes that belong to the dimensions are generated, we add another 1,000 U$[-1, 1]$ variables to the original matrix of genes. The variance-covariance matrix is a block-diagonal matrix

$$\sum = \begin{pmatrix} \sum_\rho & 0 & \cdots & \cdots & \vdots \\ 0 & \sum_\rho & 0 & \ddots & \vdots \\ \vdots & 0 & \sum_\rho & 0 & \vdots \\ \vdots & \ddots & 0 & \sum_\rho & 0 \\ \vdots & \cdots & \cdots & 0 & \sum_\rho \end{pmatrix}_{(d\times b)\times(d\times b)},$$

$$\sum_\rho = \begin{pmatrix} 1 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 1 \end{pmatrix}_{b\times b}.$$

TABLE 2
Gene Selection Results

Methods	MCL	DCL	NKI
Selected genes			
Random survival forests IFE log-rank split (w. selection)	0.34	0.44	0.29
Random survival forests IFE log-rank score split (w. selection)	0.39	0.39	0.29
Random survival forests IFE random split (w. selection)	0.34	0.45	0.29
Random survival forests IFE conserve split (w. selection)	0.35	0.42	0.29
Cforest+selection	0.29	0.43	0.29
Neural Network+selection	0.36	0.44	0.32
All genes			
Random survival forests log-rank split	0.32	0.42	0.26
Random survival forests log-rank score split	0.34	0.37	0.28
Random survival forests random split	0.39	0.41	0.28
Random survival forests conserve split	0.32	0.39	0.25
Conditional inference forests	0.34	0.4	0.23
Neural network	0.4	0.42	0.31
Support vector machine	0.42	0.42	0.26
Cox boosting	0.3	0.38	NA

Comparison of different methods with selected genes and all genes—average over 10 times (10-fold) cross validation error rates. "NA" is reported for Cox Boosting for NKI data because an R error message occurred while running this algorithm for this data set.

TABLE 3
Summary Statistics of Gene Selection

RSF IFE log-rank split	MCL	DCL	NKI
Error Rate	0.34	0.44	0.29
Genes selected (original)	3	15	14
Genes selected (training)	6 (4, 9)	10 (8, 12)	9 (9, 14)
Pairwise Correlation average	0.47	0.08	0.13
Proportions of genes selected in original set that appear in training set	33% (0, 33)	20% (13, 27)	21% (14, 29)
Cforest+selection	MCL	DCL	NKI
Error Rate	0.29	0.43	0.29
Genes selected (original)	5	29	12
Genes selected (training)	7 (4, 18)	9 (6, 12)	10 (8, 16)
Pairwise Correlation average	0.68	0.36	0.65
Proportions of genes selected in original set that appear in training set	20% (0, 60)	10% (7, 25)	25% (17, 42)
Neural Network+selection	MCL	DCL	NKI
Error Rate	0.36	0.44	0.32
Genes selected (original)	2	7	8
Genes selected (training)	3 (2, 5)	6 (4, 7)	6 (5, 8)
Pairwise Correlation average	0.77	0.10	0.12
Proportions of genes selected in original set that appear in training set	0% (0, 50)	14% (0, 28)	25% (13, 38)

Summary statistics of gene selection over the 10 times 10-fold cross validation run. Gene selected (training)—median (first quartile, third quartile). Proportions of genes—median (first quartile, third quartile).

The survival time S, is defined as $S = \exp(-\mathbf{X}\beta + \varepsilon)$, where $\mathbf{X}$ is the simulated gene expression matrix and β represents the coefficient parameter. For the null case, β equals 0 for all the predictors, $\varepsilon \sim N(0, 2)$, and the generated survival time is permuted for each run. Under the alternative case, β equals 1/2d for different numbers of informative genes, $i_g = 2, 4, 5, 6$, and 10, the ε term disappears, and $\beta = 0$ otherwise. The censoring time was generated from N(max(S),20), which yields 20-40 percent censoring for each simulated data set. We have also investigated when $\beta = 1/3d$ and $1/d; \varepsilon \sim N(0, 1)$ and N(0,3).

To calculate the error rate under the null and alternative cases, we first obtain the selected genes, gs, and calculate the 10-fold cross validation error rate and then repeat the 10-fold cross validation using a randomly selected set of genes of #gs.

5 Results and Discussions

5.1 Application to Data Sets

MCL is a data set that consists of microarray and survival data for patients with mantle cell lymphoma [38]. DCL is a data set that contains microarrays and survival information for patients with diffuse large B-cell lymphoma after chemotherapy [39]. NKI is a data set that includes microarray and survival data for patients with primary breast cancers [40].

Table 2 shows a comparison of different machine learning algorithms for survival prediction and random survival forests with gene selection (top half) or without gene selection, i.e., using all genes (bottom half). The average 10 times 10-fold cross validation error rates from the algorithms with gene selection is comparable to those using all genes. This finding is consistent with Díaz-Uriarte and Alvarez de Andrés, who investigated gene selection in random forests classification [1]. The numbers of genes selected by the different algorithms are displayed in Table 3. Table 3 also shows the number of genes selected in the different training sets, pairwise correlation among the selected genes (original), and the proportion of genes selected in the original set that appears in the training set. Unlike the $\mathrm{cforest} + \mathrm{selection}$ and neural $\mathrm{network} + \mathrm{selection}$, our iterative feature elimination algorithm with random survival forests has lower correlation among genes as it is a multivariate-based gene selection algorithm. In terms of proportions of genes in training that are in the original data, $\mathrm{cforest} + \mathrm{selection}$ and our iterative feature elimination algorithm performed similarly. Even though both $\mathrm{cforest} + \mathrm{selection}$ and neural $\mathrm{network} + \mathrm{selection}$ uses univariate Cox model as the method

TABLE 4
Impact of Standard Error

RSF IFE log-rank split Method (SE = 0)	MCL	DCL	NKI
Error Rate	0.33	0.41	0.27
Genes selected (original)	168	58	77
Genes selected (training)	55 (22, 86)	37 (24, 46)	32 (21, 50)
Proportions of genes selected in original set that appear in training set	8% (4, 11)	24% (19, 29)	17% (13, 21)
RSF IFE log-rank split Method (SE = 0.5)	MCL	DCL	NKI
Error Rate	0.34	0.43	0.28
Genes selected (original)	11	19	9
Genes selected (training)	14 (9, 22)	15 (12, 19)	14 (11, 18)
Proportions of genes selected in original set that appear in training set	0% (0, 9)	31% (26, 37)	44% (33, 56)

Impact on the average CV error rate under different standard error values for determining the mOOB.

TABLE 5
Impact of Percent of Features Dropped

RSF IFE log-rank split Method	MCL	DCL	NKI
(% Dropped=20) Error Rate	0.34	0.44	0.29
(% Dropped=50) Error Rate	0.35	0.43	0.31
(% Dropped=80) Error Rate	0.33	0.43	0.28

RSF IFE log-rank split Method (% Dropped=50)	MCL	DCL	NKI
Error Rate	0.35	0.43	0.31
Genes selected (original)	5	21	11
Genes selected (training)	5 (5, 11)	11 (11, 11)	11 (11, 21)
Proportions of genes selected in original set that appear in training set	20% (0, 20)	19% (19, 25)	27% (18, 36)
RSF IFE log-rank split Method (% Dropped=80)	MCL	DCL	NKI
Error Rate	0.33	0.43	0.28
Genes selected (original)	10	11	22
Genes selected (training)	10 (10, 10)	11 (11, 11)	22 (22, 22)
Proportions of genes selected in original set that appear in training set	0% (0, 10)	36% (27, 45)	32% (27, 36)

Impact on the average CV error rate with different percentage of features dropped.

for ranking genes, the iterative feature elimination algorithm for the former outperforms the latter in all three data sets. Integrated Brier Score, which measures how well the survival function is estimated, was also used, see Section B, available in the online supplemental material.

Table 2 shows the comparison of different split methods. Overall, none of the split criteria from random survival forests is superior to others. Given the variation in performance, unless there is a special preference, we recommend log-rank for the iterative elimination algorithm as it is the default chosen for random survival forests [16]. Table 4 shows how the gene selection algorithm performs under an alternative choice of 0 and 0.5 s.e. In terms of average CV error, the default 1 s.e. and the alternative choices of standard errors results are very similar. However, the number of genes selected are 168, 58, and 77 for MCL, DCL, and NKI, respectively, for 0 s.e. The high number of genes is much less desirable than what we found in the case of 1 s.e., with 3, 15, and 14 genes, respectively. The one standard error rule is consistent with what has been used in the machine learning literature [23], [24]. 0.5 s.e. which produces higher proportion of genes selected in the original for DCL and NKI may be considered if higher number of genes is desired. Figure of Section C, available in the online supplemental material, is a plot of the average 10 times 10-fold cross validation error rates against the number of trees used to perform the gene selection.

For all three data sets, the average cross validation error rates are consistent for 1,000 trees or more. Therefore, 1,000 trees are used in all other analyses. For the MCL data set, the error rate converges with around 500 trees. Figure of Section D, available in the online supplemental material, investigates the tradeoff between increasing correlation or strength of the trees among a forest. Overall, it shows for the three data sets that the default number of genes used for splitting performs quite well. Table 5 illustrates the impact of the percentage of features dropped on the average cross validation error rates. If we increase the percentage of features dropped from the 20 to 50 percent, the average cross validation error rates are slightly worse with 5, 21, and 11 genes chosen for MCL, DCL, and NKI, respectively. And for 80 percent dropped, it essentially removes the flexibility of the iterative feature elimination algorithm because the number of genes is fixed for all runs at 10, 11, and 22 for MCL, DCL, and NKI, respectively. Therefore, we recommend dropping 20 percent of the features at each iteration with 1,000 trees.

The selected genes for MCL were PRIM1, PCNA, and ASPM. PCNA has been shown to be associated with proliferation and survival of mantle cell lymphoma [41], [42]. The selected genes for DCL were PHKB, FLJ20703, BMP6 (2x), PTGES, TKT, PLAU, SPARC, CCL2, CXCR6, ITGAL, DAXX, CTSF, NFKBIL1, and SH3BGRL. BMP6, which was selected twice, has been identified as a potential risk predictor for diffuse large B-cell lymphoma (DLBCL) and is hypermethylated in aggressive forms of lymphoma [43]. SPARC is one of the genes in a CD5 DLBCL signature [44]. CCL2, also known as MCP-1, is produced in primary central nervous system lymphomas which consist mostly of DLBCL [45], and it is involved in the migration and localization of follicular lymphoma cells [46]. CXCR6 is one of the four chemokine receptors identified as a good classifier between MALT lymphoma and extranodal DLBCL [47]. ITGAL, also known as CD11a, is detected on some B-cell clones and is hypothesized to be related to tissue localization of B-Chronic Lymphocytic Leukemia [48]. Two research groups have discovered that DAXX is closely tied with the disease mechanism of B-cell lymphomas [49], [50]. A common NFKBIL1 variant is found to be

TABLE 6
Under the Null—No Signal

Dimensions	Genes per dimension	Average # of genes	Median # of genes	Average CV error rate
1	5	2.01	2	0.45
1	20	2.03	2	0.44
2	5	2.08	2	0.44
2	20	2.25	2	0 43
3	5	2.12	2	0.44
3	20	2.18	2	0.43
10	50	3.08	2	0.44
10	100	3.15	2	0.44

Simulation results: Average CV error rate under the null of no signal.

TABLE 7
Under the Alternative—With Informative Genes

Dimensions	Genes per dimension	Informative genes	Average # of genes	Median # of genes	Average CV error rate
1	5	2	2.76	2	0.05
1	20	5	4.99	4	0.047
1	100	10	9.31	8	0.048
2	5	4	4.34	3	0.127
2	20	4	6.12	4	0.118
2	100	4	5.64	3	0.117
3	5	6	6.56	4	0.152
3	20	6	7.37	5	0.143
3	100	6	7.55	5	0.131

Simulation results: Average CV error rate under the alternative with informative genes.

associated with non-Hodgkin lymphoma risk [51]. The selected genes for NKI are DNMT3B, FAF1, RFC4, ADFP, CIZ1, PIR, ABCF1, CANX, AP3S2, SERPINA5, CTSL2, TMEPAI, CCNB1, and HMOX1. CIZ1 is an estrogen receptor coactivator, and plays a regulatory role in the cell cycle progression of breast cancer [52], [53]. Researchers using derived PIR, PARP-inhibitor-resistant, clones were able to show that resistance to PARP inhibition can be acquired by deletion of a mutation in BRCA2 which is a well-known breast cancer gene [54]. Other researchers have found SERPINA5 to be a potential positive prognostic factor for breast cancer [55]. CCNB1 has been identified as a prognostic marker in lymph node negative breast cancer [56]. Further discussions of these genes can be found in Section E, available in the online supplemental material.

We also compared our proposed algorithm with that of univariate gene selection (single-gene-based) using Cox regression and correcting for multiple comparison using FDR [57], [58]. Under this method, the numbers of genes found to be significant at 0.05 q-value levels were 497, 7, and 897 for MCL, DCL and NKI, respectively. For the NKI data set, there were 897 found by Cox model with 5 percent FDR, and only two of the 14 found by our method did not overlap with Cox model. For the DCL data set, there were seven found by Cox model with 5 percent FDR, only two overlapping with our method. For MCL, there were 497 found by the Cox model with 5 percent FDR, and only one of the three found by our method did not overlap with the Cox model. We performed DAVID analysis to look at enriched pathways, and the DCL and NKI data sets did not result in any significantly enriched pathways after multiple testing adjustments [59]. For the MCL data set, 15 KEGG pathways were found to be significant. When we use $(\mathrm{SE} = 0)$, it identified more genes, as it chooses the iteration that has the lowest error rate. For NKI, 24 of the 77 were not found in the 897 from the univariate Cox regression model with 5 percent FDR. For DCL, 2 of the 58 were not found in the 7 from the univariate Cox regression model with 5 percent FDR. And for MCL, 107 of the 168 were not found in the 497 from the univariate Cox regression model with 5 percent FDR. With the exception of DCL, the set of genes found by using our approach is much more manageable and picks out a realistic set of genes for further testing and validation. In addition, our approach uses random forests which implicitly takes into account the way the genes interact.

Unlike univariate gene selection, our approach does not require the user to set a threshold cutoff for p-values or q-values. Our algorithm contains tuning parameters such as the s.e. that can be modified to vary the number of genes selected. We recommend using 1 s.e. as it is commonly used in the classification tree setting. Additionally, a reasonable number of genes were obtained from the three real data sets analyzed. These genes can be tested for further validation.

5.2 Simulation Results

To assess the cross validation error rate under the null of no signal and the alternative, we performed simulations as described above. For every simulated data set, we first performed the gene selection procedure and then obtained the error rates from the 10-fold cross-validation using the selected genes. For both type I error and power, we calculated the average of the 10-fold cross-validation error over 100 simulations.

From Table 6, under the case of no signal, we see that the average cross-validation error rate ranges from 0.43 to 0.45, slightly better than random. The median and average number of genes chosen is approximately two to three for various settings, including genes per dimension of 5, 20, 50, and 100 and dimensions of 1, 2, 3, and 10. For the case with informative genes, Table 7, we see that the median and average number of genes selected across the simulations is very close to the true value under different settings. The error rates are close to 0.05 for the one-dimension cases with various numbers of genes per dimension. As the number of dimensions increases to two and three, the error rates hover around 0.11 to 0.15. The above simulation demonstrates that our method is performing well under the null and alternative cases. Additional results can be found in Sections F and G, available in the online supplemental material.

5.3 Computational Time

For a comparison of the computational time of the single processor and the parallelized versions, please see Table 8. Parallelization takes place after the importance measure was calculated for the data set as a whole, therefore, only the error rate calculations were parallelized. Performance can be further improved if the parallelization can take place within the C code called within R.

TABLE 8
Computational Time

	Single version	Cluster parallel version	
	(time in minutes)		
		7 nodes	14 nodes
MCL	20	> 30	n/a
DCL	59	47	36
NKI	116	86	76

Computational time for the three real data sets in minutes

6 Conclusions

High-throughput genomics data can help predict the prognosis of patients in complex diseases like cancer. We developed a method to identify a set of prognostic genes that predicts the survival of patients, compared several machine learning methods and various node split criteria using several existing data sets, and evaluated our method via simulation. Our method performed well under the null and alternative cases. We applied our method to data sets on breast cancer, B-cell lymphoma, and mantle cell lymphoma. Using these data sets, we demonstrated the disadvantages of using a univariate gene selection method. Our approach has the advantage of being able to identify a small set of genes while preserving the predictive accuracy for survival. Univariate methods which do not take correlation into account can often pick out too many genes, making further validation difficult. However, they can be coupled with iterative feature elimination algorithms like cforest+selection developed in this paper. We also investigated the performance of our gene selection algorithm under various settings, including the percentage of features dropped, the standard error for choosing mOOB, and the number of trees used. To the best of our knowledge, our gene selection method is the first proposed to incorporate multivariate correlations in microarray data for survival outcome using machine learning methods. It will be interesting to compare the proposed method with other approaches based on gene pairs [60]. We emphasize that the goal of this procedure is not to identify signature, but to pick out genes for further validation. The algorithm proposed in this paper is for gene selection for a particular training set in high dimensional data, such as microarrays. In the future, incorporating pathways into gene selection should help enhance the predictive ability and interpretability of the findings [61].

7 Software

Three R packages geneSelRSF, pgeneSelRSF (parallelized version of geneSelRSF), and geneSelCIF are available for download at [62]: (http://www.duke.edu/~hp44/geneSelRSF.html).

Acknowledgments

Funding: National Institutes of Health (grant P01CA142538 and T32GM07205), Hong Kong RGC grant HKBU202711, Hong Kong Baptist University grant FRG1/10-11/031, and start-up funds from Duke University Medical Center.

References

[1] R. Díaz-Uriarte and S. Alvarez de Andrés, "Gene Selection and Classification of Microarray Data Using Random Forest," *BMC Bioinformatics,* vol. 7, article 3, 2006.
[2] K.B. Duan, J.C. Rajapakse, H. Wang, and F. Azuaje, "Multiple SVM-RFE for Gene Selection in Cancer Classification with Expression Data," *IEEE Trans. Nanobioscience,* vol. 4, no. 3, pp. 228-234, Sept. 2005.
[3] Y. Tang, Y.Q. Zhang, and Z. Huang, "Development of Two-Stage SVM-RFE Gene Selection Strategy for Microarray Expression Data Analysis," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 4, no. 3, pp. 365-381, July-Sept. 2007.
[4] S. Niijima and Y. Okuno, "Laplacian Linear Discriminant Analysis Approach to Unsupervised Feature Selection," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 6, no. 4, pp. 605-614, Oct.-Dec. 2009.
[5] K.Z. Mao and W. Tang, "Recursive Mahalanobis Separability Measure for Gene Subset Selection," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 8, no. 1, pp. 266-272, Jan./Feb. 2011.
[6] P.A. Mundra and J.C. Rajapakse, "SVM-RFE with MRMR Filter for Gene Selection," *IEEE Trans. Nanobioscience,* vol. 9, no. 1, pp. 31-37, Mar. 2010.
[7] L.K. Luo, D.F. Huang, L.J. Ye, Q.F. Zhou, G.F. Shao, and H. Peng, "Improving the Computational Efficiency of Recursive Cluster Elimination for Gene Selection," *IEEE/ACM Trans. Computational Biology and Bioinformatics.* vol. 8, no. 1, pp. 122-129, Jan./Feb. 2011.
[8] H.M. Bøvelstad, S. Nygård, and O. Borgan, "Survival Prediction from Clinico-Genomic Models-a Comparative Study," *BMC Bioinformatics,* vol. 10, article 413, 2009.
[9] H. Binder and M. Schumacher, "Allowing for Mandatory Covariates in Boosting Estimation of Sparse High-dimensional Survival Models," *BMC Bioinformatics,* vol. 9, article 14, 2008.
[10] K. Lee and B. Mallick, "Bayesian Methods for Variable Selection in Survival Models with Application to DNA Microarray Data," *Sankhya,* vol. 66, pp. 756-778, 2004.
[11] M. Schumacher, H. Binder, and T. Gerds, "Assessment of Survival Prediction Models Based on Microarray Data," *Bioinformatics,* vol. 23, pp. 1768-1774, 2007.
[12] W. van Wieringen, D. Kun, R. Hampel, and A. Boulesteix, "Survival Prediction Using Gene Expression Data: A Review and Comparison," *Computational Statistics and Data Analysis,* vol. 53, pp. 1590-1603, 2009.
[13] D. Dunkler, M. Schemper, and G. Heinze, "Gene Selection in Microarray Survival Studies under Possibly Non-Proportional Hazards," *Bioinformatics,* vol. 26, pp. 784-790, 2010.
[14] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.
[15] L. Breiman, "How to Use Survival Forests (SFPDV1)," http://oz.berkeley.edu/users/breiman/SF_Manual.pdf, May 2010.
[16] H. Ishwaran, U. Kogalur, E. Blackstone, and M. Lauer, "Random Survival Forests," *Annals Applied Statistics,* vol. 2, pp. 841-860, 2008.
[17] K.L. Lunetta, L.B. Hayward, J. Segal, and P. Van Eerdewegh, "Screening Large-Scale Association Study Data: Exploiting Interactions Using Random Forests," *BMC Genetics,* vol. 5, article 32, 2004.
[18] F.E. Harrell, R.M. Califf, D.B. Pryor, K.L. Lee, and R.A. Rosati, "Evaluating the Yield of Medical Tests," *J. Am. Medical Assoc.,* vol. 247, pp. 2543-2546, 1982.
[19] M. Segal, "Regression Trees for Censored Data," *Biometrics,* vol. 44, pp. 35-47, 1988.
[20] T. Hothorn and B. Lausen, "On the Exact Distribution of Maximally Selected Rank Statistics," *Computational Statistics and Data Analysis,* vol. 43, pp. 121-137, 2003.
[21] D. Naftel, E. Blackstone, and M. Turner, "Conservation of Events," unpublished, 1985.

[22] Y. Lin and Y. Jeon, "Random Forests and Adaptive Nearest Neighbors," *J. Am. Statistical Assoc.,* vol. 101, pp. 578-590, 2006.

[23] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Chapman and Hall, 1984.

[24] B. Ripley, *Pattern Recognition and Neural Networks.* Cambridge Univ. Press, 1996.

[25] C. Ambroise and G.J. McLachlan, "Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data," *Proc. Nat'l Academy of Sciences USA,* vol. 99, pp. 6562-6566, 2002.

[26] R. Simon, M. Radmacher, K. Dobbin, and L. McShane, "Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification," *J. Nat'l Cancer Inst.* vol. 95, pp. 14-18, 2003.

[27] U. Braga-Neto, R. Hashimoto, E. Dougherty, D. Nguyen, and R. Carroll, "Is Cross-Validation Better than Resubstitution for Ranking Genes?," *Bioinformatics,* vol. 20, pp. 253-258, 2004.

[28] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased Recursive Partitioning: A Conditional Inference Framework," *J. Computational Graphical Statistics,* vol. 15, pp. 651-674, 2006.

[29] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution," *BMC Bioinformatics,* vol. 8, article 25, 2007.

[30] R. Ripley, A. Harris, and L. Tarassenko, "Non-Linear Survival Analysis Using Neural Networks," *Statistics Medicine,* vol. 23, pp. 825-842, 2004.

[31] R. Ripley, "Survnnet: Feed-forward Neural Networks for Survival Analysis," R Package Version 1.1-2, 2004.

[32] T. Hothorn, P. Buhlmann, T. Kneib, M. Schmid, and B. Hofner, "Mboost: Model-Based Boosting," R Package Version 1.0-5, 2008.

[33] P. Buhlmann and T. Hothorn, "Boosting Algorithms: Regularization, Prediction and Model Fitting," *Statistical Science,* vol. 22, pp. 477-505, 2007.

[34] J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics,* vol. 29, pp. 1189-1232, 2001.

[35] P. Bühlman and B. Yu, "Boosting with the L2 Loss: Regression and Classification," *J. Am. Statistical Assoc.,* vol. 98, pp. 324-339, 2003.

[36] L. Evers, "survpack: Methods for Fitting High-Dimensional Survival Models," R Package Version 0.1-4, 2008.

[37] L. Evers and C. Messow, "Sparse Kernel Methods for High-Dimensional Survival Data," *Bioinformatics,* vol. 15, pp. 1632-1638, 2008.

[38] A. Rosenwald, G. Wright, A. Wiestner, W.C. Chan, J.M. Connors, E. Campo, R.D. Gascoyne, T.M. Grogan, H.K. Muller-Hermelink, E.B. Smeland, M. Chiorazzi, J.M. Giltnane, E.M. Hurt, H. Zhao, L. Averett, S. Henrickson, L. Yang, J. Powell, W.H. Wilson, E.S. Jaffe, R. Simon, R.D. Klausner, E. Montserrat, F. Bosch, T.C. Greiner, D.D. Weisenburger, W.G. Sanger, B.J. Dave, J.C. Lynch, J. Vose, J.O. Armitage, R.I. Fisher, T.P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, H. Holte, J. Delabie, and L.M. Staudt, "The Proliferation Gene Expression Signature is a Quantitative Integrator of Oncogenic Events that Predicts Survival in Mantle Cell Lymphoma," *Cancer Cell,* vol. 3, pp. 185-197, 2003.

[39] A. Rosenwald, G. Wright, W.C. Chan, J.M. Connors, E. Campo, R.I. Fisher, R.D. Gascoyne, H.K. Muller-Hermelink, E.B. Smeland, J.M. Giltnane, E.M. Hurt, H. Zhao, L. Averett, L. Yang, W.H. Wilson, E.S. Jaffe, R. Simon, R.D. Klausner, J. Powell, P.L. Duffey, D.L. Longo, T.C. Greiner, D.D. Weisenburger, W.G. Sanger, B.J. Dave, J.C. Lynch, J. Vose, J.O. Armitage, E. Montserrat, A. López-Guillermo, T.M. Grogan, T.P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, and L.M. Staudt, "Lymphoma/Leukemia Molecular Profiling Project: The Use of Molecular Profiling to Predict Survival after Chemotherapy for Diffuse Large-B-Cell Lymphoma," *New England J. Medicine,* vol. 346, pp. 1937-1947, 2002.

[40] M.J. van de Vijver, Y.D. He, L.J. van't Veer, H. Dai, A.A. Hart, D.W. Voskuil, G.J. Schreiber, J.L. Peterse, C. Roberts, M.J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E.T. Rutgers, S.H. Friend, and R. Bernards, "A Gene-Expression Signature as a Predictor of Survival in Breast Cancer," *New England J. Medicine,* vol. 347, pp. 1999-2009, 2002.

[41] R. Castillo, J. Mascarenhas, W. Telford, A. Chadburn, S. Friedman, and E. Schattner, "Proliferative Response of Mantle Cell Lymphoma Cells Stimulated by CD40 Ligation and IL-4," *Leukemia,* vol. 14, pp. 292-298, 2000.

[42] S. Desai, M. Maurin, M. Smith, S. Bolick, S. Dessureault, J. Tao, E. Sotomayor, and K. Wright, "PRDM1 is Required for Mantle Cell Lymphoma Response to Bortezomib," *Molecular Cancer Research,* vol. 8, pp. 907-918, 2010.

[43] M. Daibata, Y. Nemoto, K. Bandobashi, N. Kotani, M. Kuroda, M. Tsuchiya, H. Okuda, T. Takakuwa, S. Imai, T. Shuin, and H. Taguchi, "Promoter Hypermethylation of the Bone Morphogenetic Protein-6 Gene in Malignant Lymphoma," *Clinical Cancer Research,* vol. 13, pp. 3528-3535, 2007.

[44] M. Suguro, H. Tagawa, Y. Kagami, M. Okamoto, K. Ohshima, H. Shiku, Y. Morishima, S. Nakamura, and M. Seto, "Expression Profiling Analysis of the CD5+ Diffuse Large B-Cell Lymphoma Subgroup: Development of a CD5 Signature," *Cancer Science,* vol. 97, pp. 868-874, 2006.

[45] R. Kitai, K. Ishisaka, K. Sato, T. Sakuma, T. Yamauchi, Y. Imamura, H. Matsumoto, and T. Kubota, "Primary Central Nervous System Lymphoma Secretes Monocyte Chemoattractant Protein 1," *Medical Molecular Morphology,* vol. 40, pp. 18-22, 2007.

[46] H. Husson, E. Carideo, A. Cardoso, S. Lugli, D. Neuberg, O. Munoz, L. de Leval, J. Schultze, and A. Freedman, "MCP-1 Modulates Chemotaxis by Follicular Lymphoma Cells," *British J. Haematology,* vol. 115, pp. 554-562, 2001.

[47] A. Deutsch, A. Aigelsreiter, E. Steinbauer, M. Frühwirth, H. Kerl, C. Beham-Schmid, H. Schaider, and P. Neumeister, "Distinct Signatures of B-Cell Homeostatic and Activation-Dependent Chemokine Receptors in the Development and Progression of Extragastric MALT Lymphomas," *J. Pathology,* vol. 215, pp. 431-444, 2008.

[48] E. Kimby, J. Rincon, M. Patarroyo, and H. Mellsted, "Expression of Adhesion Molecules CD11/CD18 (Leu-CAMs, β2-integrins), CD54 (ICAM-1) and CD58 (LFA-3) in B-Chronic Lymphocytic Leukemia," *Leukemia Lymphoma,* vol. 13, p. 297, 1994.

[49] R. Perlman, W. Schiemann, M. Brooks, H. Lodish, and R. Weinberg, "TGF-Beta-Induced Apoptosis is Mediated by the Adapter Protein Daxx that Facilitates JNK Activation," *Nature Cell Biology,* vol. 3, pp. 708-714, 2001.

[50] A. Brieger, S. Boehrer, S. Schaaf, D. Nowak, M. Ruthardt, S. Kim, P. Atadja, D. Hoelzer, P. Mitrou, E. Weidmann, and K. Chow, "In bcr-abl-Positive Myeloid Cells Resistant to Conventional Chemotherapeutic Agents, Expression of Par-4 Increases Sensitivity to Imatinib (STI571) and Histone Deacetylase-Inhibitors," *Biochemical Pharmacology,* vol. 68, pp. 85-93, 2004.

[51] S.S. Wang, M.P. Purdue, J.R. Cerhan, T. Zheng, I. Menashe, B.K. Armstrong, Q. Lan, P. Hartge, A. Kricker, Y. Zhang, L.M. Morton, C.M. Vajdic, T.R. Holford, R.K. Severson, A. Grulich, B.P. Leaderer, S. Davis, W. Cozen, M. Yeager, S.J. Chanock, N. Chatterjee, and N. Rothman, "Common Gene Variants in the Tumor Necrosis Factor (TNF) and TNF Receptor Superfamilies and NF-kB Transcription Factors and Non-Hodgkin Lymphoma Risk," *PLoS One,* vol. 4, no. 4, p. e5360, Apr. 2009.

[52] P. den Hollander, S. Rayala, D. Coverley, and R. Kumar, "Ciz1, a Novel DNA-Binding Coactivator of the Estrogen Receptor Alpha, Confers Hypersensitivity to Estrogen Action," *Cancer Research,* vol. 66, pp. 11021-11029, 2006.

[53] P. den Hollander and R. Kumar, "Dynein Light Chain 1 Contributes to Cell Cycle Progression by Increasing Cyclin-dependent Kinase 2 Activity in Estrogen-Stimulated Cells," *Cancer Research,* vol. 66, pp. 5941-5949, 2006.

[54] S. Edwards, R. Brough, C. Lord, R. Natrajan, R. Vatcheva, D. Levine, J. Boyd, J. Reis-Filho, and A. Ashworth, "Resistance to Therapy Caused by Intragenic Deletion in BRCA2," *Nature,* vol. 451, pp. 1111-1115, 2008.

[55] R. Castelló, J. Landete, F. España, C. Vázquez, C. Fuster, S. Almenar, L. Ramón, K. Radtke, and A. Estellés, "Expression of Plasminogen Activator Inhibitors Type 1 and Type 3 and Urokinase Plasminogen Activator Protein and mRNA in Breast Cancer," *Thrombosis Research,* vol. 120, pp. 753-762, 2007.

[56] E. Niméus-Malmström, A. Koliadi, C. Ahlin, M. Holmqvist, L. Holmberg, R. Amini, K. Jirström, F. Wärnberg, C. Blomqvist, M. Fernö, and M. Fjällskog, "Cyclin B1 is a Prognostic Proliferation Marker with a High Reproducibility in a Population-Based Lymph Node Negative Breast Cancer Cohort," *Int'l J. Cancer,* vol. 127, pp. 961-967, 2010.

[57] D. Cox, "Regression Models and Life-Tables," *J. the Royal Statistical Soc. Series B,* vol. 34, pp. 187-220, 1972.

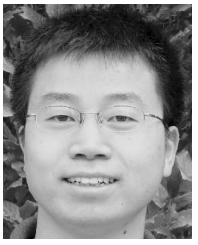
[58] J. Storey and R. Tibshirani, "Statistical Significance for Genome-wide Studies," *Proc. Nat'l Academy of Sciences USA,* vol. 100, pp. 9440-9445, 2003.

[59] D. Huang, B. Sherman, and R. Lempicki, "Systematic and Integrative Analysis of Large Gene Lists Using DAVID Bioinformatics Resources," *Nature Protocols,* vol. 4, pp. 44-57, 2009.
[60] E. Motakis, A. Ivshina, and V. Kuznetsov, "Data-Driven Approach to Predict Survival of Cancer Patients," *IEEE Eng. in Medicine and Biology Magazine,* vol. 28, no. 4, pp. 58-66, July/Aug. 2009.
[61] H. Pang, D. Datta, and H. Zhao, "Pathway Analysis using Random Forests with Bivariate Node-Split for Survival Outcomes," *Bioinformatics* vol. 26, pp. 250-258, 2010.
[62] R Development Core Team "R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing," Vienna, Austria, http://www.R-project.org, 2010.

Herbert Pang received the BA degree in mathematics and computer science from the University of Oxford, England, and the PhD degree in biostatistics from Yale University. Currently, he is working as an assistant professor in the Department of Biostatistics and Bioinformatics at Duke University. His main research interests include bioinformatics, genomics, and clinical trials.

Stephen L. George received the BA degree in mathematics from Texas Tech University, the MES degree in statistics from North Carolina State University, and the PhD degree in statistics from Southern Methodist University. Currently, he is working as a professor in the Department of Biostatistics and Bioinformatics at Duke University. His research interests include statistical methodology for the design and analysis of clinical trials. He is a fellow of American Statistical Association, a fellow and former president of the Society of Clinical Trials, a former Executive Committee member of the International Biometric Society, and a former member of the FDA Oncology Drugs Advisory Committee.

Ken Hui received the BS and MS degrees in applied mathematics and biology from Yale University. Currently, he is working toward the MD/PhD degree at Yale University School of Medicine.



Tiejun Tong received the BS degree in electronic engineering and MS degree in statistics from the University of Science and Technology of China, and the PhD degree in statistics from the University of California at Santa Barbara. Currently, he is working as an assistant professor in the Department of Mathematics at the Hong Kong Baptist University. His main research interests include high-dimensional data analysis and shrinkage estimation.