Analytica Chimica Acta 740 (2012) 20–26

Contents lists available at SciVerse ScienceDirect

Analytica Chimica Acta

journal homepage: www.elsevier.com/locate/aca

Random frog: An efficient reversible jump Markov Chain Monte Carlo-like approach for variable selection with applications to gene selection and disease classification

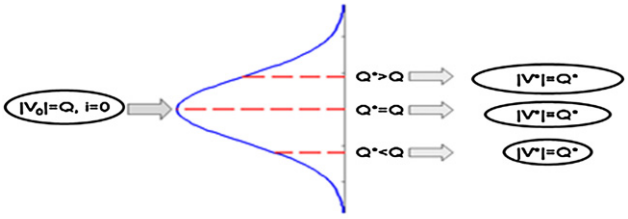
Hong-Dong Li[a,b], Qing-Song Xu[a,b], Yi-Zeng Liang[a,*]

[a] College of Chemistry and Chemical Engineering, Central South University, Changsha 410083, PR China
[b] School of Mathematic Sciences, Central South University, Changsha 410083, PR China

HIGHLIGHTS

- The proposed method possesses advantages of RJMCMC algorithms.
- The proposed method is easier to implement than RJMCMC.
- Competitive results over published work were obtained.
- Random frog is computationally very efficient.

GRAPHICAL ABSTRACT



Using a normal distribution to generate a random number Q* that guides jumping Between different models.

ARTICLE INFO

Article history:
Received 6 March 2012
Received in revised form 13 June 2012
Accepted 19 June 2012
Available online 28 June 2012

Keywords:
Variable selection
Gene expression-based disease classification
Markov Chain Monte Carlo
Random frog

ABSTRACT

The identification of disease-relevant genes represents a challenge in microarray-based disease diagnosis where the sample size is often limited. Among established methods, reversible jump Markov Chain Monte Carlo (RJMCMC) methods have proven to be quite promising for variable selection. However, the design and application of an RJMCMC algorithm requires, for example, special criteria for prior distributions. Also, the simulation from joint posterior distributions of models is computationally extensive, and may even be mathematically intractable. These disadvantages may limit the applications of RJMCMC algorithms. Therefore, the development of algorithms that possess the advantages of RJMCMC methods and are also efficient and easy to follow for selecting disease-associated genes is required. Here we report a RJMCMC-like method, called random frog that possesses the advantages of RJMCMC methods and is much easier to implement. Using the colon and the estrogen gene expression datasets, we show that random frog is effective in identifying discriminating genes. The top 2 ranked genes for colon and estrogen are Z50753, U00968, and Y10871_at, Z22536_at, respectively. (The source codes with GNU General Public License Version 2.0 are freely available to non-commercial users at: http://code.google.com/p/randomfrog/.)

1. Introduction

The developed microarray experiment that monitors expression levels of thousands of genes involved in different disease phenotypes, has been gaining extensive applications for cancer classification, and also becoming commonly used in the fields of biomedical and clinical research [1–4]. The main goal of cancer classification includes: predicting prognosis, proposing therapy according to the clinical situation, advancing therapeutic studies, etc. Thus, it is of interest for physicians and clinicians to establish the rules for accurate tumor classification before the administration of any treatment to the patient, in order to avoid unnecessary treatment and/or propose the most appropriate therapies. Definitely, cancer classification plays a key role in cancer treatment [1], and studies of gene expression data based cancer classification have been widely reported [5–10].

* Corresponding author. Tel.: +86 731 88830831; fax: +86 731 8830831.
E-mail address: yizeng_liang@263.net (Y.-Z. Liang).

0003-2670/$ – see front matter © 2012 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.aca.2012.06.031

One key point in cancer classification is to identify a small number of relevant genes that can be used for accurate prediction of the phenotype in a new individual [11]. However, the identification of these genes is a great challenge in the context of genomic study, since not only gene expression data are usually of a very high dimensionality but also the number of samples available is comparatively rather small. As is known, it is a "large p, small n" problem [12,13]. To our knowledge, some methods have been proposed that could be used to identify genes that are potentially responsible for cancer classification, *e.g.*, class distinction correlation [1], support vector machines (SVM) [14,15], sparse logistic regression [16], entropy-based method [17] and Bayesian approaches like reversible jump Markov Chain Monte Carlo (RJMCMC) [18,19].

Among these established methods, the RJMCMC method is in theory a very powerful one and it has been gaining some successful applications. Through the implementation of fixed-dimensional and trans-dimensional searching among different models, RJMCMC constructs a MCMC chain in the model space in which each model is characterized by a model index and its associated model parameters. The resulting MCMC chain will converge into the joint distribution of model indices and model-specific parameters if the transition probability satisfies the detailed balance condition, which says: the transition probability that a general sate A in the chain moves to a general state B is the same as that when A and B are reversed. In spite of its advantages, the applications of RJMCMC are still limited by several factors; for example, to design an effective RJMCMC algorithm, prior distributions over the number of variables and model parameters have to be appropriately assigned so that posterior distributions of interested parameters will be computationally tractable. In addition, constructing a MCMC chain is usually computationally extensive. In practice, these two factors indeed limit the use and applications of RJMCMC.

Borrowing the merits of RJMCMC techniques and aiming to establish a mathematically simple and computationally efficient method, here we report a RJMCMC-like algorithm called random frog for gene selection. Like any RJMCMC method, random frog realizes a search in the model space through both fixed-dimensional and trans-dimensional moves between different models. By doing so, a pseudo-MCMC chain can be computed and then used to calculate, for each gene, a selection probability that measures genes' relevance, and this can be used as a gene selection criterion. The performance of random frog was tested on two benchmark datasets. Our results show that random frog is computationally efficient and can single out a small number of genes that lead to significant improvement in terms of prediction errors over the previously reported results.

2. Theory and algorithm

2.1. Reversible jump MCMC

The reversible jump MCMC algorithm was firstly introduced by Green [18], and since then, different versions of RJMCMC algorithms have been reported [19,20]. Here we only aim to present the algorithm in a schematic form, as described in reference [20]. The current model of Markov chain is assumed to be denoted as (k, θ_k), where k stands for the model index and θ_k refers to the corresponding model parameter. Then, the RJMCMC algorithm can be briefly described as follows:

Step 1. Propose a visit to model M_{k^*} with probability $J(k \to k^*)$. Here k and k^* denote different model indices.
Step 2. Generate a new model θ_{k^*}.
Step 3. Calculate the acceptance probability of the new model as the product of model ratio and proposal ratio.

Looping these three steps generates a sample $(k_i, i = 1, \ldots, N)$ for the model indicators and the posterior probability of the kth model given data $(\mathbf{X}, \mathbf{y})$ can be estimated using the following formula,

$$P(k|X, y) = \frac{1}{L}\sum_{i=1}^{N} I_k(k_i) \quad (1)$$

where $I_k(\cdot) = 1$ if $k = k_i$ and zero otherwise.

2.2. Random frog

Borrowing the framework of reversible jump MCMC, in the present work we developed the random frog algorithm. Let $\mathbf{X}$, of size $n \times p$, denote the design matrix consisting of n samples in rows and p variables in columns and $\mathbf{y}$, the class label vector of size $n \times 1$, with elements equal to 1 or −1 in a binary classification case. As previously mentioned, random frog is a general strategy for variable selection. So, for a method to build a classification model, this needs to be specified in order to implement random frog. In the present work, we chose partial least squares-linear discriminant analysis (PLS-LDA) to construct classifiers. The rationales behind our choice were: (1) PLS-LDA is quite strong in handling highly correlated data, such as gene expression data, and (2) a PLS-LDA model is linear and thus easy to interpret.

Random frog is a method that works in an iterative manner. The schematic is shown in Fig. 1. Briefly, random frog works in three steps: (1) a variable subset $\mathbf{V}_0$ containing Q variables is initialized randomly, (2) propose a candidate variable subset $\mathbf{V}^*$ including Q^* variables based on $\mathbf{V}_0$, accept $\mathbf{V}^*$ with a certain probability as $\mathbf{V}_1$, and replace $\mathbf{V}_0$ using $\mathbf{V}_1$. This step is looped until N iterations are finished. (3) Finally compute a selection probability of each variable which can be used as a measure of variable importance.

The algorithm of random frog will be fully detailed in the following paragraphs.

2.2.1. Parameters and initialization of random frog

There are five tuning parameters controlling the performance of random frog. These parameters as well as their meanings are listed below.

(1) N: the number of iterations. This is a very important factor. It needs to be sufficiently large to achieve convergence. According to our experiences, $N = 10{,}000$ can often work well.
(2) Q: the number of variables contained in the initialized variable set. Q can be any number of between 1 and p. In our experiments, it was found that Q had influence on the iterative behavior only at the beginning but had no significant effects on the performance of random frog. Usually, a number smaller than p, can work.
(3) θ: a factor controlling the variance of a normal distribution from which the number of variables to enter a candidate variable subset is sampled. Details on this parameter will be given in Section 2.2. By default, we set θ to 0.3.
(4) ω: a factor whose value should be larger than 1. Details on this parameter will be given in Section 2.2. By default, we set ω to 3.
(5) η: a parameter that ranges from 0 to 1. This is the upper bound of the probability for accepting a candidate variable subset $\mathbf{V}^*$ whose performance is not better than $\mathbf{V}_0$. By default, it is set to 0.1.

Before running the random frog, these five parameters are initialized. A variable subset $\mathbf{V}_0$, consisting of Q variables selected randomly, is initialized. Denote the set containing all p variables as $\mathbf{V}$. Thus, we have $\mathbf{V}_0 \subset \mathrm{V}$.

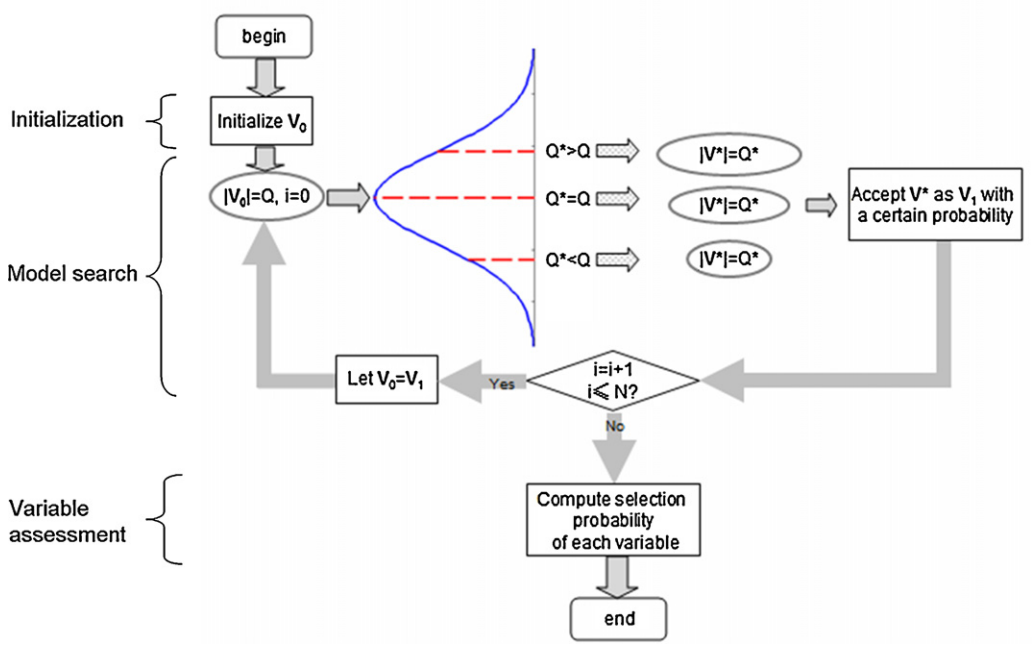


Fig. 1. Flowchart of the random frog algorithm. Given an initial variable subset $\mathbf{V}_0$ with its cardinality denoted by $|\mathbf{V}_0| = Q$, a random number is generated from the normal distribution with mean Q and standard deviation θQ, where R is a predefined positive number, *e.g.* 0.3. This random number is then rounded to its nearest integer, denoted by Q^*. Based on $\mathbf{V}_0$, a candidate variable subset V^* that contains Q^* variables is generated. Next, accept $\mathbf{V}^*$ as $\mathbf{V}_1$ with a certain probability. Finally, let $\mathbf{V}_0 = \mathbf{V}_1$ and repeat the above procedures until N iterations are finished.

2.2.2. Probability-guided model searching in random frog

At first, a random number is generated from a normal distribution Norm(Q, θQ), where Q and θQ are the mean and standard deviation of this distribution, respectively. This random number is rounded to its nearest integer, denoted as Q^*. Q^* is the number of variables of the candidate variable subset $\mathbf{V}^*$. The reason why we choose θQ as the standard deviation is that it can adjust the number of variables included in a model automatically and efficiently. For example, given $\theta = 0.3$, if the current model includes 500 variables, *e.g.* $Q = 500$, the dimension of the candidate model Q^* can then vary in a very wide range from Norm(500, 150), which allows for great jumps between differently dimensioned models. In contrast, if the current model is low dimensional with $Q = 10$, Q^* can only be valued from the distribution Norm(10, 3), which means the change in the number of variables is limited, allowing for the refinement of model dimensionality.

Once Q^* is determined, the next step is to propose a candidate variable subset $\mathbf{V}^*$ that contains exactly Q^* variables. There are three possible situations:

(1) If $Q^* = Q$, let $\mathbf{V}^* = \mathbf{V}_0$. This is the easiest case to handle.
(2) If $Q^* < Q$, a PLS-LDA model is first built using $\mathbf{V}_0$ and the regression coefficient of each variable in this model is recorded and compared. The $Q - Q^*$ variables associated with the smallest absolute regression coefficients are removed from $\mathbf{V}_0$. The remaining Q^* variables form a candidate subset $\mathbf{V}^*$. In this case, selective deletion of variables from $\mathbf{V}_0$ is realized.
(3) If $Q^* > Q$, a variable subset $\mathbf{T}$ with $\omega(Q^* - Q)$ variables sampled from $\mathbf{V} - \mathbf{V}_0$ randomly is generated. A PLS-LDA model is established using the combination of $\mathbf{V}_0$ and $\mathbf{T}$. The Q^* variables with the largest absolute regression coefficients in this PLS-LDA model are retained and collected as a candidate subset $\mathbf{V}^*$. In this situation, selective deletion of variables from $\mathbf{V}_0$ and/or addition of variables from $\mathbf{T}$ to $\mathbf{V}_0$ would happen.

Briefly, the use of the proposed normal distribution to control number of variables implicitly introduces the operation of variable addition and deletion, which provides the basis of model searching in a general model space. After the candidate variable subset $\mathbf{V}^*$ is obtained, the next step is to determine whether $\mathbf{V}^*$ can be accepted. First we compute cross validated misclassification errors using $\mathbf{V}_0$ and $\mathbf{V}^*$, respectively. Denote them as Err and Err*. If $Err^* \leq Err$, accept $\mathbf{V}^*$ as $\mathbf{V}_1$. Otherwise accept $\mathbf{V}^*$ as $\mathbf{V}_1$ with probability $\eta Err/Err^*$. Clearly, $Err/Err^* < 1$, so $\eta Err/Err^* < \eta$. This is why η is called the upper bound of the probability for accepting $\mathbf{V}^*$ as discussed before.

Finally, $\mathbf{V}_0$ is updated using the variables in $\mathbf{V}_1$ and this iteration is repeated until N loops are finished.

2.2.3. Compute selection probability of each variable

After N iterations, in total N variable subsets can be obtained. Denote the frequency for the jth variable, $j = 1, 2, \ldots, p$, to be selected in these N variable subsets as N_j. Then, for each variable, its selection probability can be calculated using formula (2).

$$Prob_j = \frac{N_j}{N}, \quad j = 1, 2, \ldots, p \tag{2}$$

As can be expected, the more important a variable is, the more likely it is to be selected into these N variable subsets. So, this selection probability can serve as a measure of variable importance and, therefore, can be used as an index for variable selection. Also, building on the analysis of a large number of sub-models sampled from the model space, random frog can be viewed as an implementation of model population analysis (MPA), which is a general framework for designing data analysis algorithms [21–24].

By the way, one may want to determine the optimal number of variables that should be included in a classification model. One way is to build a series of PLS models with increasing number of variables until a predefined maximal number is achieved followed by computing the classification error of each model using cross validation. The optimal number of variables can then be chosen as the one that is associated with the lowest error.

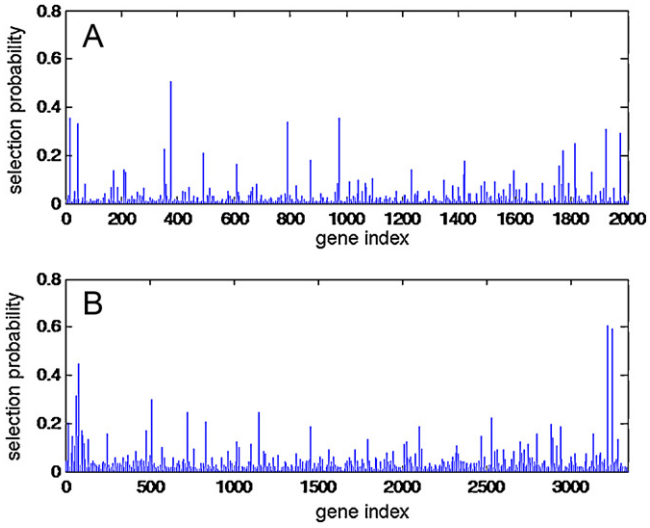


Fig. 2. The selection probability of each gene averaged over 20 runs of random frog for the colon data at $Q = 50$ (A) and estrogen data (B) at $Q = 100$.

3. Results and discussion

3.1. Colon data

This dataset contains the gene expression profiles measured in 40 tumor and 22 normal colon tissues for 6500 human genes obtained by applying the Affymetrix gene chip technology. A screening of 2000 genes with the highest minimal intensity across the samples has been made by Alon et al. [25] and is freely available at http://microarray.princeton.edu/oncology/. The analysis of this dataset has been carried out in several reported studies, see references [26–28].

As discussed before, there are five parameters, N, Q, θ, ω and η, that need to be set before running random frog. As can be expected, the larger the N is, the more likely the random frog method is to identify the best variable subset. However, a too large N value will result in a high computational cost. Considering both performances and computational cost, N was set to 10,000 in the present study. With respect to Q, a series of Q values, *i.e.* [2, 10, 30, 50, 100], were tested, from which an optimal Q value is chosen using cross validation. The other three parameters, θ, ω and η, do not have significant influences on the results and were set to 3, 0.3 and 0.1, respectively, by default. Before running random frog to perform gene selection, each gene was standardized to have zero mean and unit variance.

As can be expected, the selection probability of genes cannot be reproduced exactly as a consequence of the embedded Monte Carlo strategy in random frog. To reduce the influence of this random factor, we therefore run random frog 20 times at each Q value and take the average over these 20 runs as the criterion for assessing the importance of each gene. For illustration, the average selection probability at $Q = 50$ is shown in Fig. 2A. From this plot, it can be found that only a small portion of genes exhibit a relatively high selection probability, whereas the selection probability of most genes is very low, even close to zero. This finding might be an indication that the number of disease-related genes is small, thus providing computational evidence supporting the necessity of gene selection for disease classification.

To build a model that is predictive of clinical outcome, a subset of genes needs to be selected. For each Q value, we first rank all the genes based on their selection probability. Then, to comprehensively investigate the influence of the number of genes included in the model as well as to seek an optimal number of genes, nine gene sets, which consist of the highest ranked 10, 25, 50, 75, 100 and 200, were considered here. To make our results comparable with published studies, we first used leave-one-out cross validation (LOOCV) to evaluate performances of these 6 gene sets. However, it is known that LOOCV usually gives over-optimistic results. So the repeated double Cross Validation (rdCV) developed by Varmuza et al. [29] was also employed for performance evaluation because this method can give much better estimation of prediction errors. The misclassification errors using both LOOCV and rdCV at different Q values are displayed in Fig. 3A and B respectively. As can be seen, these misclassification error curves with different Q values show the similar changing trend, suggesting that the number of genes in the initialized subset does not affect much the predictive performances. For both LOOCV and rdCV, these misclassification error curves first decrease and then keep stable or go up slightly with increased genes, implying gene selection can greatly improve classification accuracies of a predictive model. It is also found that prediction errors from rdCV are consistently higher than LOOCV, confirming that rdCV is a better estimator of model performances.

Table 1
The top ranked 15 genes for the colon cancer data.

ID	Gene ID	Gene description
1	Z50753	*H. sapiens* mRNA for GCAP II/uroguanylin precursor[a,b]
2	U00968	Sterol regulatory element binding protein 1 (human)
3	H20709	Myosin light chain alkali, smooth-muscle isoform (human)[b]
4	R88740	ATP synthase coupling factor 6, mitochondrial precursor (human)[b]
5	T57619	40S ribosomal protein S6 (*Nicotiana tabacum*)
6	H64807	Placental folate transporter (*Homo sapiens*)[b]
7	K03474	Human Mullerian inhibiting substance gene, complete cds
8	X93510	*H. sapiens* mRNA for 37 kDa LIM domain protein
9	T57882	Myosin heavy chain, nonmuscle Type A (*Homo sapiens*)[b]
10	H08393	Collagen alpha 2(XI) chain (*Homo sapiens*)[a,b]
11	R87126	Myosin heavy chain, nonmuscle (*Gallus gallus*)[a,b]
12	T41207	Come operon protein 3 (*Bacillus subtilis*)
13	J02854	Myosin regulator light chain 2, smooth muscle isoform (human)[a,b]
14	L06895	*Homo sapiens* antagonizer of myc transcriptional activity (Mad) mRNA, complete cds
15	H16096	Mitochondrial processing protease beta subunit precursor (*Rattus norvegicus*)

[a] Ben-Dor et al. [26].
[b] Yang and Song [30].

By comparison, the optimal Q value was chosen to be 50 in terms of classification accuracies. The 15 most significant genes ranked by the selection probability at $Q = 50$ are presented in Table 1. Eight of them were found to be reported previously by Ben-Dor et al. [26] and Yang and Song [30]. The highest ranked gene is uroguanylin precursor Z50753. In the work of Notterman et al. [31], it was shown that a reduction of uroguanylin might be an indication of colon tumors. In addition, it was reported by Shailubhai et al. [32] that treatment using uroguanylin exhibits a therapeutic effect to the reduction in pre-cancerous colon ploys. The gene R87126 (myosin heavy chain, non-muscle) is also of interest. Yam et al. [33] reported that its isoform B can work as a tumor suppressor and is also known for its role in the cytoskeletal network. Indeed, it is very challenging to perform biological validation of each gene. However, these literature reports provide evidence showing the meaningfulness of the identified significant genes.

For comparison, predictive results of classification models reported in the literature are summarized here. Furey et al. [14] misclassified 6 samples using SVM based on LOOCV, leading to a prediction error = 0.097. In Nguyen and Rocke [34], their best result

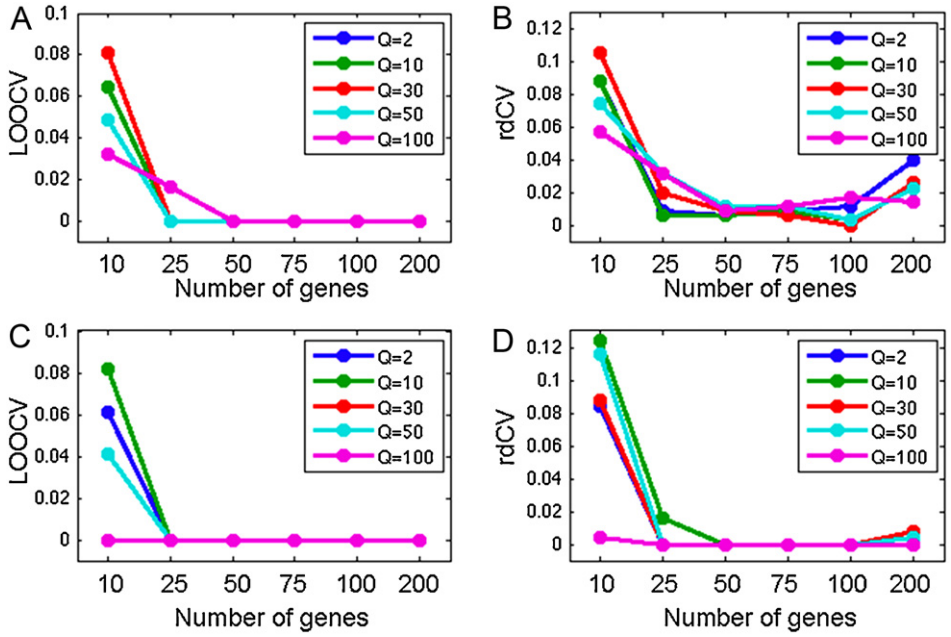


Fig. 3. The misclassification errors resulting from LOOCV as well as rdCV against the number of genes included in a PLS-LDA model at different Q values for colon (A and B) and estrogen (C and D) data, respectively.

was obtained using PLS including 50 or 100 genes with the misclassification error 0.065. Using the top ranked 200 genes based on Wilcoxon rank test, Dettling and Buhlmann [27] achieves the misclassification errors as shown: 0.145 (LogitBoost, optimal), 0.177 (Adaboost), 0.177 (1-nearest-neighbor) and 0.145 (CART). Pochet et al. [15] shows a mean classification error 0.180 by employing SVM-based methods. In the work of Ma and Huang [4], the regularized ROC method based mean misclassification error is 0.140 with a standard deviation 0.070. In contrast, the LOOCV error of PLS-LDA models using genes selected by random frog achieves zero, suggesting significant improvement over the reported results.

In addition, we also compared our method with another three methods: *t*-statistic, mutual information and the sequential forward selection (SFS) motivated method [35]. We employed rdCV for comparison. The rdCV misclassification errors resulting from these three methods as well as the random frog on 6 gene subsets with a maximum of 200 genes allowed are presented in Table 2. Clearly, the prediction errors of random frog are lower than the other three methods. Furthermore, to examine whether selected genes could better reflect between-class pattern differences, a principal component analysis (PCA) model was constructed on both the original data and the reduced data containing only the highest ranked 25 genes, respectively. Comparing the scores plots in Fig. 4A and B, it can be found that a much better separation was achieved using only selected genes, indicating that these genes associated with high selection probability have a predictive value. Summing up, we have reasons to say that the random frog is very powerful in terms of lower prediction errors as well as a smaller number of gene subsets, and therefore can be considered as a promising alternative for gene selection.

3.2. Estrogen data

This dataset was firstly presented by West et al. [36] and Spang et al. [37], consisting of the gene expression values of 49 breast tumor samples on 7129 genes. Out of these samples, 25 are LN+ and the remaining 24 are LN−. After pretreating this data, following the same methods described in Ma and Huang [4], 3333 genes in total were kept. The raw data is freely available at http://mgm.duke.edu/genome/dna_micro/work/.

For the tuning parameters, *N*, *Q*, θ, ω and η, we used the same setting as for the colon data. *N* was set to 10,000, a series of *Q* values, *i.e.* [2, 10, 30, 50, 100] were tested, and default values of θ, ω and η used. Before analysis using random frog, each gene was standardized to have zero mean and unit variance. By analogy with the analysis of colon data, the selection probability averaged over 20 runs of random frog was used to assess the importance of each gene. Taking $Q = 100$ as an example, the average selection probability is shown in Fig. 2B. For this dataset, we also found that only a small number of genes displayed a relatively high selection probability, again, suggesting that most of these measured genes are not relevant to the disease under investigation and therefore the necessity of gene selection.

Similarly, to study the influence of the number of genes included in PLS-LDA classifiers as well as to determine an optimal number of genes, twelve gene sets, which consist of the highest ranked 10, 25, 50, 75, 100 and 200 genes, were considered. The resulting LOOCV and rdCV misclassification errors corresponding to different *Q* values are shown in Fig. 3C and D as a function of the number of genes. The similar changing trend of these misclassification error curves, again, provides evidences that the initialization of the random frog algorithm has little effect on the performance. Of note, only 10 genes identified with $Q = 100$ achieved a misclassification error zero, showing the great potential of random frog for gene selection of high dimensional data. By comparison, the optimal *Q* value for this data was chosen to be 100 since $Q = 100$ lead to the minimum error with the least number of genes. By analogy, the 15 most significant genes ranked by the selection probability at $Q = 100$ are presented in Table 3. To our knowledge, these genes have not been reported to be associated with the lymph nodal status. It could be interesting to further investigate whether and how these genes may be interlinked to breast cancer.

For this dataset, Dettling and Buhlmann [27] yields classification errors of 0.020 (LogitBoost, optimal), 0.06 (AdaBoost, 100 iterations) and 0.040 (CART) using 100 genes. Using a regularized ROC method, Ma and Huang [4] achieved a mean misclassification error of 0.060 with a 0.070 standard deviation. Compared to these

Table 2
Comparison of the repeated double cross validated misclassification errors using different variable selection methods.

		10	25	50	75	100	200
Colon							
A	*t*-Statistic	15.86	13.71	14.71	13.14	13.57	16.43
B	Mutual information	11.57	11.71	12.71	14.71	14.86	16.43
C	SFS-motivated method	14.57	7.71	3.14	0.57	1.14	2.29
D	Random frog	8.86	1.71	0.00	0.57	0.29	2.29
Estrogen							
A	*t*-Statistic	4.00	7.60	8.80	9.20	6.80	6.80
B	Mutual information	14.40	14.40	16.00	16.40	14.40	13.20
C	SFS-motivated method	14.00	2.80	0.80	0.00	0.00	0.00
D	Random frog	6.80	0.00	0.00	0.00	0.00	0.00

Table 3
The top ranked 15 genes for the estrogen data.

ID	Gene ID	Gene description
1	Y10871_at	*H. sapiens* twist gene
2	Z22536_at	*Homo sapiens* ALK-4 mRNA, complete CDS
3	AFFX-CreX-3_st	Bacteriophage P1 cre recombinase protein
4	AFFX-BioB-3_at	*E. coli* bioB gene biotin synthetase
5	HG2280-HT2376_at	D-Amino-acid oxidase
6	J02982_f_at	Human glycophorin B mRNA, complete cds
7	M13686_s_at	Human pulmonary surfactant-associated protein mRNA, complete cds
8	AFFX-BioB-M_at	*E. coli* bioB gene biotin synthetase
9	U88898_r_at	Human endogenous retroviral H protease/integrase-derived ORF1 mRNA
10	L04733_at	*Homo sapiens* kinesin light chain mRNA, complete cds
11	AB002318_at	Human mRNA for KIAA0320 gene, partial cds
12	X66087_at	*H. sapiens* a-myb mRNA
13	U37408_at	Human CtBP mRNA, complete cds
14	X72012_at	*H. sapiens* end mRNA for endoglin
15	M61916_at	Human laminin B1 chain mRNA, complete cds

results, the proposed random frog method performs the best in terms of misclassification errors. We also used rdCV to compare our method with *t*-statistic, mutual information and the sequential forward selection (SFS) motivated method [35]. The results are given in Table 2. The best performance is also achieved by random frog. Furthermore, we constructed principal component analysis (PCA) models using the original data and the reduced data with the highest ranked 25 genes included. The scores plots are shown in Fig. 4C and D, respectively. Of notice, the two classes of samples based on all 3333 genes completely overlap with each other with no separation observed. In contrast, full separation was achieved between these two phenotypes of samples when using only 25 genes identified by random frog, implying that these 25 genes might be possible biomarkers that are of diagnostic value when working collectively. To conclude, the random frog method has been shown to be of great power for variable selection of high dimensional data.

3.3. Computational time

For both the colon and estrogen data, one run of random frog with 10,000 iterations takes only around 6.2 min in MATLAB 7.1

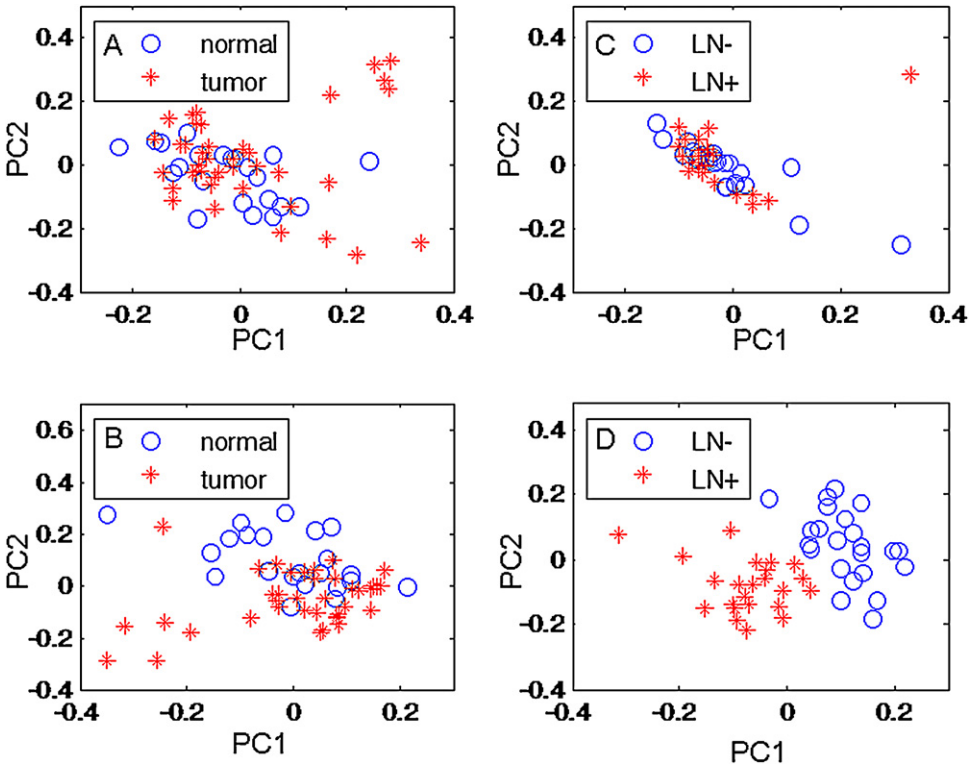


Fig. 4. Plot A and C shows the principal component plot before and after gene selection of the colon data, respectively. The plots for the estrogen data are given in Plot B and D, respectively.

in a Mac with Intel core 2 1.6 GHz CPU and 2 G RAM. As a rough reference, in Yang and Song's work [30], 258 min are needed to run a MCMC chain with 60,000 iterations for the colon data in a PC with Intel Core2 1.86 GHz CPU 1 GB RAM.

4. Conclusions

Identifying a small subset of genes that can be used for accurate prediction of the clinical outcome of a new individual is of great value. However, the task of searching such a subset out of thousands of genes represents a great challenge. In the present work, a RJMCMC-like method, called random frog, was developed, conducting a search in a model space through the realization of both fixed-dimensional and trans-dimensional jumps between different models. The advantage of random frog is that no demanding mathematical formulation is needed and no prior distributions need to be specified like in formal RJMCMC methods, which makes it easier to implement and computationally very efficient. Experimental results on two freely available microarray datasets showed that random frog is capable of identifying a small number of genes with discriminating power and that the classifiers established using these genes outperform those previously reported in terms of classification accuracies. The proposed random frog method is expected to gain more applications in the future.

Acknowledgements

This work was financially supported by the National Nature Foundation Committee of P.R. China (Grants No. 20875104, No. 21075138 and No. 21105129), the Graduate degree thesis Innovation Foundation of Central South University (CX2010B057) and the Fundamental Research Funds for the Central Universities (2011QNZT053). The studies meet with the approval of the university's review board. The study is approved by the review board of Central South University. Specifically, the authors would like to acknowledge Miguel Ferro in University of Barcelona for his great help in the language.

References

[1] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Science 286 (1999) 531–537.
[2] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Nat. Med. 7 (2001) 673–679.
[3] S.M. Dhanasekaran, T.R. Barrette, D. Ghosh, R. Shah, S. Varambally, K. Kurachi, K.J. Pienta, M.A. Rubin, A.M. Chinnaiyan, Nature 412 (2001) 822–826.
[4] S. Ma, J. Huang, Bioinformatics 21 (2005) 4356–4362.
[5] L. Zhang, W. Zhou, V.E. Velculescu, S.E. Kern, R.H. Hruban, S.R. Hamilton, B. Vogelstein, K.W. Kinzler, Science 276 (1997) 1268–1272.
[6] Y.-J. Lu, D. Williamson, J. Clark, R. Wang, N. Tiffin, L. Skelton, T. Gordon, R. Williams, B. Allan, A. Jackman, C. Cooper, K. Pritchard-Jones, J. Shipley, Proc. Natl. Acad. Sci. U.S.A. 98 (2001) 9197–9202.
[7] C. Virtanen, Y. Ishikawa, D. Honjoh, M. Kimura, M. Shimane, T. Miyoshi, H. Nomura, M.H. Jones, Proc. Natl. Acad. Sci. U.S.A. 99 (2002) 12357–12362.
[8] P. Qiu, Z.J. Wang, K.J.R. Liu, Z.-Z. Hu, C.H. Wu, Bioinformatics 23 (2007) 198–206.
[9] L. Shen, M. Toyota, Y. Kondo, E. Lin, L. Zhang, Y. Guo, N.S. Hernandez, X. Chen, S. Ahmed, K. Konishi, S.R. Hamilton, J.-P.J. Issa, Proc. Natl. Acad. Sci. U.S.A. 104 (2007) 18654–18659.
[10] L. Wang, C. Aliferis, BMC Bioinformatics 9 (2008) 319.
[11] K.Y. Yeung, R.E. Bumgarner, A.E. Raftery, Bioinformatics 21 (2005) 2394–2402.
[12] H. Zou, T. Hastie, J. R. Statist. Soc. B 67 (2005) 301–320.
[13] E. Candes, T. Tao, Ann. Statist. 35 (2007) 2313–2351.
[14] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Bioinformatics 16 (2000) 906–914.
[15] N. Pochet, F. De Smet, J.A.K. Suykens, B.L.R. De Moor, Bioinformatics 20 (2004) 3185–3195.
[16] G.C. Cawley, N.L.C. Talbot, Bioinformatics 22 (2006) 2348–2355.
[17] X. Liu, A. Krishnan, A. Mondry, BMC Bioinformatics 6 (2005) 76.
[18] P.J. Green, Biometrika 82 (1995) 711–732.
[19] M.G. Tadesse, N. Sha, M. Vannucci, J. Am. Stat. Assoc. 100 (2005) 602–617.
[20] H.F. Lopes, A Note on Reversible Jump Markov Chain Monte Carlo, Graduate School of Business, The University of Chicago, 2006, February 1st.
[21] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, J. Chemometr. 24 (2009) 418–423.
[22] H.-D. Li, M.-M. Zeng, B.-B. Tan, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, Metabolomics 6 (2010) 353–361.
[23] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, B.-B. Tan, B.-C. Deng, C.-C. Lin, IEEE/ACM Trans. Comput. Biol. 8 (2011) 1633–1641.
[24] H.-D. Li, Y.-Z. Liang, Q.-S. Xu, D.-S. Cao, TrAC (2012), http://dx.doi.org/10.1016/j.trac.2011.11.007.
[25] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Proc. Natl. Acad. Sci. U.S.A. 96 (1999) 6745–6750.
[26] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, Z. Yakhini, J. Comput. Biol. 7 (2000) 559–584.
[27] M. Dettling, P. Buhlmann, Bioinformatics 19 (2003) 1061–1069.
[28] W. Ma, X. Zhang, F. Luan, H. Zhang, R. Zhang, M. Liu, Z. Hu, B.T. Fan, J. Phys. Chem. A 109 (2005) 3485–3492.
[29] P. Filzmoser, B. Liebmann, K. Varmuza, J. Chemometr. 23 (2009) 160–171.
[30] A.-J. Yang, X.-Y. Song, Bioinformatics 26 (2010) 215–222.
[31] D.A. Notterman, U. Alon, A.J. Sierk, A.J. Levine, Cancer Res. 61 (2001) 3124–3130.
[32] K. Shailubhai, H. Yu, K. Karunanandaa, J. Wang, S. Eber, Y. Wang, N. Joo, H. Kim, B. Miedema, S. Abbas, S. Boddupalli, M. Currie, L. Forte, Cancer Res. 60 (2000) 5151–5157.
[33] J.W.P. Yam, K.W. Chan, W.-L.W. Hsiao, Oncogene 20 (2001) 58–68.
[34] D. Nguyen, D.M. Rocke, Bioinformatics 18 (2002) 39–50.
[35] O. Gualdrón, J. Brezmes, E. Llobet, A. Amari, X. Vilanova, B. Bouchikhi, X. Correig, Sens. Actuators B: Chem. 122 (2007) 259–268.
[36] M. West, C. Blanchette, H. Dressmna, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, J. Nevins, Proc. Natl. Acad. Sci. U.S.A. 98 (2001) 11462–11467.
[37] R. Spang, C. Blanchette, H. Zuzan, J. Marks, J. Nevins, M. West, Proceedings of the German Conference on Bioinformatics GCB, 2001.