PERGAMON

Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

Journal of the Franklin Institute 341 (2004) 137–156

Journal
of The
Franklin Institute

www.elsevier.com/locate/jfranklin

A Bayesian approach to nonlinear probit gene selection and classification

Xiaobo Zhou[a,b], Xiaodong Wang[c,*], Edward R. Dougherty[a,d]

[a] *Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA*

[b] *Harvard University Medical School and Brigham and Womens Hospital, Goldsen Building 524, 220 Longwood Avenue, Boston, MA 02115, USA*

[c] *Department of Electrical Engineering, Columbia University, 717 Schapiro CEPSR 500 West 120th Street, New York, NY 10027, USA*

[d] *Department of Pathology, University of Texas M.D. Anderson Cancer Center, Houston, TX 77030, USA*

Abstract

We consider the problem of gene selection and classification based on the expression data. Specifically, we propose a bootstrap Bayesian gene selection method for nonlinear probit regression. A binomial probit regression model with data augmentation is used to transform the binomial problem into a sequence of smoothing problems. The probit regressor is approximated as a nonlinear combination of the genes. A Gibbs sampler is employed to find the strongest genes. Some numerical techniques to speed up the computation are discussed. We then develop a nonlinear probit Bayesian classifier consisting of a linear term plus a nonlinear term, the parameters of which are estimated using the sequential Monte Carlo technique. These new methods are applied to analyze several data sets, including the hereditary breast cancer data, the small round blue-cell tumor data, and the acute leukemia tumor data. The experimental results show the proposed methods can effectively find important genes which are consistent with the existing biological belief, and the classification accuracies are very high. Some robustness and sensitivity properties of the proposed methods are also discussed to deal with noisy microarray data.

Keywords: Gene microarray; Cancer classification; Bayesian gene selection; Gibbs sampler; Sequential Monte Carlo

*Corresponding author. Tel.: + 1-212-854-6592; fax: + 1-212-932-9421.
E-mail address: wangx@ee.columbia.edu (X. Wang).

0016-0032/$30.00 © 2004 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.jfranklin.2003.12.010

1. Introduction

The availability of cDNA microarrays makes it possible to measure simultaneously the expression levels for thousands of genes [1]. In addition to the enormous scientific potential of DNA microarrays to help understanding gene regulation and interactions [2–4], microarrays have very important applications in pharmaceutical and clinical research. Since Golub et al. [1] proposed a weighted voting scheme for molecular classification of acute leukemia, many existing machine learning methods have been applied to gene classification problems [5–8]. In particular, given the thousands of genes and the small amount of data samples, gene selection becomes a very important issue. In the past decade, a number of variable (or gene) selection methods have been proposed, e.g., the support vector machine method [6], the genetic algorithm [8,9], the perceptron method [10], the Bayesian variable selection [7,11,12], and the minimum description length principle for model selection [13]. The leave-one-out classification error is usually adopted as a measure in gene classification [7,14]. All the above methods do not consider the selection bias [15] in gene extraction because gene selection is not performed in training the different models at each stage of the leave-one-out process [14]. To address this selection bias problem, in this paper, we propose a bootstrap Bayesian gene selection for nonlinear probit regression consisting of a linear term plus a nonlinear term. We use a binomial regression model (probit regressor) with data augmentation to transform the binomial problem into a sequence of smoothing problems. The probit regressor is approximated as a nonlinear combination of the genes. A Gibbs sampler is employed to find the strongest genes. Some numerical techniques to speed up the computation are also discussed.

After the strongest genes are selected, we need to estimate the model based on the selected genes. We again use the same binomial probit regression model with data augmentation to turn the binomial problem into a sequence of smoothing problems. Linear probit regression model can usually be estimated using the Gibbs sampler [7,16]. However, for the nonlinear probit regression model, experimental results show that the performance of this method is limited. Recently the powerful sequential Monte Carlo (SMC) technique for numerical Bayesian computation has been employed to solve a variety of problems [17–21]. Many real-world signal processing problems involve nonlinearity and nonGaussianity. Consequently, it is usually not possible to derive the exact solutions based on standard criteria such as maximum likelihood, maximum a posteriori probability or minimum mean-squared error. Classical sub-optimal methods, such as the extended Kalman filter and the Gaussian sum approximation are easy to implement, but they cannot handle severe nonlinearity and therefore often give poor results [19]. Here, we propose to use the SMC method to estimate the nonlinear probit classifier for gene classification.

The remainder of this paper is organized as follows. In Section 2, we propose the bootstrap Bayesian gene selection for the nonlinear probit regression. In Section 3, we propose a nonlinear probit classifier based on the SMC estimation method. Section 4 provides some experimental results. Section 5 concludes the paper.

2. Gene selection

2.1. Problem formulation

Let $\boldsymbol{w} = [w_1, \ldots, w_n]^{\mathrm{T}}$ denote the class labels, where $w_i = 0$ indicates sample i being cancer 1, and $w_i = 1$ indicates sample i being cancer 2 or no cancer, for $i = 1, 2, \ldots, n$. Assume there are p genes and n samples. Denote x_{ij} as the measurement of the expression level of the jth gene for the ith sample where $j = 1, 2, \ldots, p$. Define the gene expression matrix as

$$\boldsymbol{X} = \begin{bmatrix} \text{Gene 1} & \text{Gene 2} & \ldots & \text{Gene}\, p \\ x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}. \tag{1}$$

Due to the small sample size, here we adopt a probit regression model composed of a linear term plus a nonlinear term. Then w_i and the gene expression levels $\boldsymbol{x}_i \triangleq [x_{i,1}, x_{i,2}, \ldots, x_{i,p}]$ are related through [16]

$$P(w_i = 1 \mid \boldsymbol{x}_i) = \Phi\left(\sum_{j=1}^{p} a_j x_{ij} + \sum_{k=1}^{K} b_k \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_k)\right), \tag{2}$$

with

$$\phi(\boldsymbol{x}_i, \boldsymbol{\mu}_k) \triangleq \exp\{-\lambda_k \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2\}, \tag{3}$$

where $\phi(\cdot)$ is a radial basis function; $\|\cdot\|$ denotes a distance metric (usually Euclidean or Mahalanobis); $\boldsymbol{\beta} \triangleq [a_1, a_2, \ldots, a_p, b_1, \ldots, b_K]^{\mathrm{T}}$ contains the parameters and Φ is the standard normal cumulative distribution function; $\boldsymbol{\mu}_k = [\mu_{k,1}, \ldots, \mu_{k,m}]$ contains the centers of the K clusters, whose values are obtained by using the fuzzy C-means clustering algorithm [12]. The parameters $\{\lambda_k\}_{k=1}^{K}$ are empirically set as 1.0. In this study, we fix $K = 2$. The motivation to setting K centers by using clustering is that the different sample sets of some genes may have different distributions because we consider two different cancer sample sets in this study. Define the following n independent latent variable $y_1, \ldots, y_n$:

$$y_i = \sum_{j=1}^{p} a_j x_{ij} + \sum_{k=1}^{K} b_k \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_k) + e_i, \quad i = 1, \ldots, n, \tag{4}$$

where $e_i \sim \mathcal{N}(0, 1)$. Denote $\boldsymbol{y} \triangleq [y_1, \ldots, y_n]^{\mathrm{T}}$, $\boldsymbol{e} \triangleq [e_1, \ldots, e_p]^{\mathrm{T}}$ and

$$\boldsymbol{D} \triangleq \begin{bmatrix} x_{1,1} & \ldots & x_{1,p} & \phi(\boldsymbol{x}_1, \boldsymbol{\mu}_1) & \ldots & \phi(\boldsymbol{x}_1, \boldsymbol{\mu}_K) \\ x_{2,1} & \ldots & x_{2,p} & \phi(\boldsymbol{x}_2, \boldsymbol{\mu}_1) & \ldots & \phi(\boldsymbol{x}_2, \boldsymbol{\mu}_K) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,p} & \phi(\boldsymbol{x}_n, \boldsymbol{\mu}_1) & \ldots & \phi(\boldsymbol{x}_n, \boldsymbol{\mu}_K) \end{bmatrix}. \tag{5}$$

Then Eq. (4) can be expressed in a vector form as

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\beta} + \boldsymbol{e}, \tag{6}$$

where $\boldsymbol{e} \triangleq [e_1, \ldots, e_n] \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$. Define $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_{p+K}]$ as the $(p+K) \times 1$ indicator vector with the jth element γ_j such that $\gamma_j = 0$ if $\beta_j = 0$ (the variable is not selected) and $\gamma_j = 1$ if $\beta_j \neq 0$ (the variable is selected). Given $\boldsymbol{\gamma}$, let $\boldsymbol{\beta}_\gamma$ consist of all nonzero elements of $\boldsymbol{\beta}$ and let $\boldsymbol{X}_\gamma$ be the columns of $\boldsymbol{X}$ corresponding to those of $\boldsymbol{\gamma}$ that are equal to 1. In this study, for simplicity, we fix the nonlinear term and only consider gene selection. Hence $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_p, 1, 1]$, i.e., $\gamma_{p+1} = 1$ and $\gamma_{p+2} = 1$.

2.2. Bootstrap Bayesian gene selection

A Gibbs sampler is employed to estimate all the parameters. Given $\boldsymbol{\gamma}$, the prior distribution of $\boldsymbol{\beta}_\gamma$ is $\boldsymbol{\beta}_\gamma \sim \mathcal{N}(\boldsymbol{0}, c(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1})$, where c is a constant (we set $c = 100$ [7] in this study). Since the detailed derivations of the posterior distributions of the parameters are similar to those in [7], here we simply summarize the procedure for Bayesian variable selection. Denote

$$S(\boldsymbol{\gamma}, \boldsymbol{y}) \triangleq \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \frac{c}{c+1}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}_\gamma(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{y}. \tag{7}$$

Then the Gibbs sampling algorithm for estimating $\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{y}$ is as follows:

- Draw $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma} \mid \boldsymbol{y})$, where $p(\boldsymbol{\gamma} \mid \boldsymbol{y}) \propto (1+c)^{-p_\gamma/2}\exp[-\frac{1}{2}S(\boldsymbol{\gamma}, \boldsymbol{y})]\prod_{j=1}^{p+2}\pi_j^{\gamma_j}(1-\pi_j)^{1-\gamma_j}$. Here $p_\gamma = \sum_{j=1}^{p+2}\gamma_j$ and $\pi_j = P(\gamma_j = 1)$ is a prior probability to select the jth gene. This parameter is often set as a small number due to small sample size. If π_j is chosen in a bigger value, then we found that often times $(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}$ does not exist. We usually sample each γ_j independently from
$$p(\gamma_j \mid \boldsymbol{y}, \boldsymbol{\gamma}_{\neq j}) \propto (1+c)^{-p_\gamma/2}\exp[-\tfrac{1}{2}S(\boldsymbol{\gamma}, \boldsymbol{y})]\pi_j^{\gamma_j}(1-\pi_j)^{1-\gamma_j}, \quad j = 1, \ldots, p, \tag{8}$$
where $\boldsymbol{\gamma}_{\neq j} \triangleq (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_p, 1, 1)$.
- Draw $\boldsymbol{\beta}$ from $p(\boldsymbol{\beta} \mid \boldsymbol{\gamma}, \boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{V}_\gamma\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{y}, \boldsymbol{V}_\gamma)$, where $\boldsymbol{V}_\gamma = c/(1+c)(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}$.
- Draw y_i, $i = 1, \ldots, n$ from a truncated normal distribution as follows [22]: $p(y_i \mid \boldsymbol{\beta}, w_i = 1) \propto \mathcal{N}(\boldsymbol{X}_i\boldsymbol{\beta}, 1)1_{\{y_i > 0\}}$ and $p(y_i \mid \boldsymbol{\beta}, w_i = 0) \propto \mathcal{N}(\boldsymbol{X}_i\boldsymbol{\beta}, 1)1_{\{y_i < 0\}}$.

In this study, 35,000 Gibbs iterations are implemented with the first 5000 as the burn-in period. Then we obtain the Monte Carlo samples as $\{\boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{y}^{(t)}, t = 1, \ldots, T\}$, where $T = 35{,}000$. Finally, we count the number of times that each gene appears in $\{\boldsymbol{\gamma}^{(t)}, t = 5001, \ldots, T\}$. The genes with the highest appearance frequencies play the strongest role in predicting the target gene. Note that in the first step of the above iteration, we need to sample $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma}|\boldsymbol{y}, \boldsymbol{\beta})$ according to standard Gibbs sampling technique, however the information of $\boldsymbol{\beta}$ is actually included in $S(\boldsymbol{\gamma}, \boldsymbol{y})$, so we can sample $\boldsymbol{\gamma}$ directly from $p(\boldsymbol{\gamma} \mid \boldsymbol{y})$. See the details in [7,23].

Here we use bootstrap technique [24] to make the Bayesian gene selection robust. In practice, the sample size n is typically small. Let $\boldsymbol{z}$ denote the vector of p gene variables. Denote $\boldsymbol{Z} = [\boldsymbol{z}(1), \boldsymbol{z}(2), \ldots, \boldsymbol{z}(n)]$ as n realizations (i.e., samples) of $\boldsymbol{z}$. At each iteration of the bootstrap procedure, n random draws are performed on $\boldsymbol{Z}$ to form a "resample" $\boldsymbol{Z}^* = [\boldsymbol{z}^*(1), \boldsymbol{z}^*(2), \ldots, \boldsymbol{z}^*(n)]$. Then we perform the above

Bayesian gene selection procedure and obtain the frequency of each genes. Finally, the bootstrap method for estimating the frequency of each gene is the mean of the resamples. Here we employ the balanced bootstrap [25,26], which is an alternative sampling method that can increase the precision of the bootstrap bias. The balanced bootstrap forces each observation to occur Q times in the Q bootstrap samples. Balanced bootstrap samples can be generated by constructing a population with Q copies of each of the n observations, then randomly permuting that population. $Q = 25$ bootstrap replications are used in this study. Moreover, we fix the original data set as one of the Q resamples.

2.3. Some implementation issues

The computational complexity of the Bayesian gene selection algorithm in the previous section is very high. For example, if there are 3000 gene variables, then for each iteration, we have to compute the matrix inverse $(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}$ 3000 times because we need to compute (8) for each gene. Hence some fast algorithms are employed to deal with the problem.

Computation of $p(\gamma_j \mid \boldsymbol{y}, \boldsymbol{\gamma}_{\neq j})$ *in Eq.* (8): Because γ_j only takes 0 or 1, we can take a closer look at $p(\gamma_j = 1 \mid \boldsymbol{y}, \boldsymbol{\gamma}_{\neq j})$ and $p(\gamma_j = 0 \mid \boldsymbol{y}, \boldsymbol{\gamma}_{\neq j})$. Let $\boldsymbol{\gamma}^1 = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_j = 1, \gamma_{j+1}, \ldots, \gamma_p, 1, 1)$ and $\boldsymbol{\gamma}^0 = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_j = 0, \gamma_{j+1}, \ldots, \gamma_p, 1, 1)$. After straightforward computation of Eq. (8), we have $p(\gamma_j = 1 \mid \boldsymbol{y}, \boldsymbol{\gamma}_{\neq j}) \propto 1/(1+h)$ with $h = (1-\pi_j)/\pi_j \exp\{S(\boldsymbol{\gamma}^1, \boldsymbol{y}) - S(\boldsymbol{\gamma}^0, \boldsymbol{y})/2\}\sqrt{1+c}$. If $\boldsymbol{\gamma} = \boldsymbol{\gamma}^0$ before γ_j is generated, that means we have obtained $S(\boldsymbol{\gamma}^0, \boldsymbol{y})$, then we only need to compute $S(\boldsymbol{\gamma}^1, \boldsymbol{y})$, and vice versa.

Fast computation of $S(\boldsymbol{\gamma}, \boldsymbol{y})$ *in Eq.* (7): From the above discussion, it is crucial to compute $S(\boldsymbol{\gamma}, \boldsymbol{y})$ fast when a gene variable is added or removed from $\boldsymbol{\gamma}$. Denote

$$E(\boldsymbol{\gamma}, \boldsymbol{y}) = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}_\gamma(\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{X}_\gamma)^{-1}\boldsymbol{X}_\gamma^{\mathrm{T}}\boldsymbol{y}. \tag{9}$$

Then Eq. (9) can be computed using the fast QR decomposition, QR-delete and QR-insert algorithms when a variable is added or removed [27, Chapter 10.1.1b]. Now we want to estimate $S(\boldsymbol{\gamma}, \boldsymbol{y})$ in Eq. (7). Comparing Eqs. (9) and (7), after straightforward computation, $S(\boldsymbol{\gamma}, \boldsymbol{y})$ is given by

$$S(\boldsymbol{\gamma}, \boldsymbol{y}) = \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + cE(\boldsymbol{\gamma}, \boldsymbol{y})}{1+c}. \tag{10}$$

Thus after computing $E(\boldsymbol{\gamma}, \boldsymbol{y})$ using QR decomposition, QR-delete and QR-insert algorithms, we then obtain $S(\boldsymbol{\gamma}, \boldsymbol{y})$. The computation complexity can be significantly reduced compared with the original algorithm [7] due to the above techniques.

3. Sequential Monte Carlo estimation for probit model

After obtaining the strongest genes, say m genes $x_1, x_2, \ldots, x_m$, now we can focus on the nonlinear probit classifier estimate. The nonlinear probit model is still given by model (2). The difference is that here all parameters $[a_1, \ldots, a_m, b_1, b_2]$ and $\boldsymbol{\mu}_1$ and

$\boldsymbol{\mu}_2$ are unknown. The SMC method [17–21] is a powerful tool to solve the nonlinear problem in Eq. (2). Rewrite Eq. (2) as

$$y_t = \boldsymbol{a}_t\boldsymbol{x}_t + \sum_{k=1}^{2} b_{k,t}\phi(\boldsymbol{x}_t, \boldsymbol{\mu}_{k,t}) + e_t \triangleq \boldsymbol{D}_t\boldsymbol{\alpha}_t + e_t, \quad t = 1, \ldots, n, \tag{11}$$

where $\boldsymbol{a}_t = [a_{t,1}, \ldots, a_{t,m}]$; $\boldsymbol{x}_t = [x_{t,1}, \ldots, x_{t,m}]$; $\boldsymbol{D}_t = [\boldsymbol{x}_t, \phi(\boldsymbol{x}_t, \boldsymbol{\mu}_{t,1}), \phi(\boldsymbol{x}_t, \boldsymbol{\mu}_{t,2})]$, $\boldsymbol{\alpha}_t = [\boldsymbol{a}_t, b_{t,1}, b_{t,2}]^{\mathrm{T}}$ and $e_t \sim \mathcal{N}(0, 1)$. Note that we need to estimate the parameters in Eq. (11) based on $\boldsymbol{D}_t$ and y_t for $t = 1, \ldots, n$. A state-space representation is adopted to describe the parameters over time:

$$\boldsymbol{\mu}_{k,t+1} = \boldsymbol{\mu}_{k,t} + \epsilon_\mu, \quad k = 1, 2, \tag{12}$$

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \epsilon_\alpha, \tag{13}$$

where $\epsilon_\mu \sim \mathcal{N}(0, \delta_\mu^2)$ and $\epsilon_\alpha \sim \mathcal{N}(0, \delta_\alpha^2 \boldsymbol{I}_{m+2})$ [28]. In this study, we set $\delta_\mu^2 = 0.001$ and $\delta_\alpha^2 = 0.001$. For notational simplicity, denote $\boldsymbol{A}_t \triangleq \{\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_t\}$, $\boldsymbol{\mu}_t \triangleq [\boldsymbol{\mu}_{t,1}, \boldsymbol{\mu}_{t,2}]$, $\boldsymbol{B}_t \triangleq \{\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_t\}$, $\boldsymbol{\theta}_t \triangleq \{\boldsymbol{\alpha}_t, \boldsymbol{\mu}_t\}$, $\boldsymbol{\Theta}_t \triangleq \{\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_t\}$, $\boldsymbol{X}_t \triangleq \{x_1, \ldots, x_t\}$, and $\boldsymbol{Y}_t \triangleq \{y_1, \ldots, y_t\}$. The joint distribution of the proposed model with $\boldsymbol{\Theta}_t$ is given by

$$\begin{aligned} p(\Theta_t, \boldsymbol{X}_t, \boldsymbol{Y}_t) &\propto p(\boldsymbol{Y}_t \mid \Theta_t, \boldsymbol{X}_t) p(\boldsymbol{A}_t) p(\boldsymbol{B}_t) \\ &= \left[\prod_{l=1}^{t} p(y_l \mid \boldsymbol{\alpha}_l, \boldsymbol{\mu}_l, \boldsymbol{x}_l) p(\boldsymbol{\alpha}_l \mid \boldsymbol{\alpha}_{l-1}) p(\boldsymbol{\mu}_l \mid \boldsymbol{\mu}_{l-1})\right] p(\boldsymbol{\alpha}_0) p(\boldsymbol{\mu}_0). \end{aligned} \tag{14}$$

Multivariable normal distributions are adopted to represent the priors $p(\boldsymbol{\alpha}_0)$ and $p(\boldsymbol{\mu}_0)$. Our aim is to estimate the joint posterior distribution $p(\boldsymbol{\Theta}_t \mid \boldsymbol{X}_t, \boldsymbol{Y}_t)$. However, it is impossible to derive closed-form analytical expressions for this distribution. As a result, we resort to the SMC method. Next, we first summarize the SMC method, and then apply it to estimate the probit model in Eq. (11).

3.1. SMC principle

Consider the following dynamic system modelled in a state-space form:

$$\begin{aligned} &\text{state equation} \quad z_t = f_t(z_{t-1}, \boldsymbol{u}_t), \\ &\text{observation equation} \quad \boldsymbol{y}_t = g_t(z_t, \boldsymbol{v}_t), \end{aligned} \tag{15}$$

where $z_t, \boldsymbol{y}_t, \boldsymbol{u}_t$ and $\boldsymbol{v}_t$ are, respectively, the state variable, the observation, the state noise, and the observation noise at time t. Let $\boldsymbol{Z}_t = (z_0, z_1, \ldots, z_t)$ and $\boldsymbol{Y}_t = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Suppose an on-line inference of $\boldsymbol{Z}_t$ is of interest; that is, at current time t we wish to make a timely estimate of a function of the state variable $\boldsymbol{Z}_t$, say $h(\boldsymbol{Z}_t)$, based on the currently available observation, $\boldsymbol{Y}_t$. With the Bayes theorem, we realize that the optimal solution to this problem is $E\{h(\boldsymbol{Z}_t) \mid \boldsymbol{Y}_t\} = \int h(\boldsymbol{Z}_t) p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)\, \mathrm{d}\boldsymbol{Z}_t$. In most cases, an exact evaluation of this expectation is analytically intractable because of the complexity of such a dynamic system. SMC methods, which are based on importance sampling, give us viable choices to the required estimation. The basic idea of SMC is to draw m random samples $\{\boldsymbol{Z}_t^{(j)}\}_{j=1}^{N}$ from some *trial* distribution $q(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$, if it is difficult to draw the samples directly from the target distribution

$p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$. By associating the weight

$$w_t^{(j)} = \frac{p(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)}{q(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)} \tag{16}$$

to the sample $\boldsymbol{Z}_t^{(j)}$, we can approximate the quantity of interest, $E\{h(\boldsymbol{Z}_t) \mid \boldsymbol{Y}_t\}$, as

$$E_p\{h(\boldsymbol{Z}_t) \mid \boldsymbol{Y}_t\} \cong \frac{1}{W_t} \sum_{j=1}^{N} h(\boldsymbol{Z}_t^{(j)}) w_t^{(j)} \tag{17}$$

with $W_t \triangleq \sum_{j=1}^{m} w_t^{(j)}$. The pair $(\boldsymbol{Z}_t^{(j)}, w_t^{(j)})$, $j = 1, \ldots, N$, is called a *properly weighted sample* with respect to the target distribution $p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$.

To implement an online estimation of the posterior density, a set of random samples properly weighted with respect to $p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$ are needed for any time t. A Markovian structure of the state equation allows us to implement a recursive importance sampling strategy. Suppose a set of properly weighted samples $\{(\boldsymbol{Z}_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^{N}$ with respect to $p(\boldsymbol{Z}_{t-1} \mid \boldsymbol{Y}_{t-1})$ are available at time $(t-1)$. Then a set of a properly weighted samples, $\{\boldsymbol{Z}_t^{(j)}, w_t^{(j)}\}_{j=1}^{N}$ with respect to $p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$ at time t are given by the following procedure [17–21]. For $j = 1, \ldots, N$:

- Draw a sample $z_t^{(j)}$ from a trial distribution $q(z_t \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)$ and let $\boldsymbol{Z}_t^{(j)} = (\boldsymbol{Z}_{t-1}^{(j)}, z_t^{(j)})$;
- Compute the importance weight

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{p(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)}{p(\boldsymbol{Z}_{t-1}^{(j)} \mid \boldsymbol{Y}_{t-1}) q(z_t^{(j)} \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)}.$$

The algorithm is initialized by drawing a set of i.i.d. samples $z_0^{(1)}, \ldots, z_0^{(N)}$ from $p(z_0 \mid \boldsymbol{y}_0)$, where $\boldsymbol{y}_0$ represents the "null" information, and $p(z_0 \mid \boldsymbol{y}_0)$ corresponds to the prior distribution of z_0. If there is prior information for z_0, then it is set according to the prior information.

3.2. SMC for probit regression

Note that the full conditional distribution of y_t in Eq. (11) is truncated normal [7,16]: $(y_t \mid \boldsymbol{\alpha}_t, w_t = 1) \sim \mathcal{N}(\boldsymbol{D}_t \boldsymbol{\alpha}_t, 1)$ truncated left by 0, and $(y_t \mid \boldsymbol{\alpha}, w_t = 0) \sim \mathcal{N}(\boldsymbol{D}_t \boldsymbol{\alpha}_t, 1)$ truncated right by 0. We next summarize our SMC algorithm for estimating the above nonlinear probit regression model as the following three steps:

(1) Bayesian importance sampling step
   - For $j = 1, \ldots, N$, sample $\boldsymbol{\theta}_t^{(j)} \sim q(\boldsymbol{\theta}_t \mid \boldsymbol{\Theta}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t)$, and set $\boldsymbol{\Theta}_t^{(j)} \triangleq (\boldsymbol{\Theta}_{t-1}^{(j)}, \boldsymbol{\theta}_t^{(j)})$.
   - For $j = 1, \ldots, N$, evaluate the importance weights:

$$w_t^{(j)} = w_{t-1}^{(j)} \frac{p(\boldsymbol{\Theta}_t^{(j)} \mid \boldsymbol{X}_t, \boldsymbol{Y}_t)}{p(\boldsymbol{\Theta}_{t-1}^{(j)} \mid \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) q(\boldsymbol{\theta}_t^{(j)} \mid \boldsymbol{\Theta}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t)}.$$

(2) Resampling: multiple samples $\boldsymbol{\Theta}_t^{(j)}$ with the normalized importance weights $w_t^{(j)}$, to obtain N random samples $\boldsymbol{\Theta}_t^{(j)}$. See Section 3.4.
(3) For $j = 1, \ldots, N$, draw $y_t^{(j)}$. (Actually, the above $\boldsymbol{Y}_t$ should be replaced by $\boldsymbol{Y}_t^{(j)} \triangleq \{y_1^{(j)}, \ldots, y_t^{(j)}\}$, however, for notational simplicity, we neglect (j) for $\boldsymbol{Y}_t$.)

We next describe the above steps separately.

3.3. Sampling step

Firstly, note that the parameters $\boldsymbol{\alpha}_t$ are dependent on $\boldsymbol{\mu}_t$, hence next we only consider to sample $\boldsymbol{\mu}_t$ and the weight w_t in the first step of SMC. In order to implement the SMC, we need to obtain a set of Monte Carlo samples of the parameters $\{(\boldsymbol{\mu}_t^{(j)}, w_t^{(j)})\}_{j=1}^N$, properly weighted with respect to $p(\boldsymbol{B}_t \mid \boldsymbol{Y}_t)$. In the prediction stage, samples are obtained as follows: $(\boldsymbol{\mu}_t \mid \boldsymbol{B}_{t-1}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) \sim p(\boldsymbol{\mu}_t \mid \boldsymbol{B}_{t-1})$ (here we also assume $(\boldsymbol{\mu}_t \mid \boldsymbol{B}_{t-1}, \boldsymbol{X}_t, \boldsymbol{Y}_{t-1}) \sim p(\boldsymbol{\mu}_t \mid \boldsymbol{B}_{t-1})$) [28]. Apply the Bayes' rule, the posterior at time t is

$$\begin{aligned}
p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{X}_t, \boldsymbol{Y}_t) &\propto p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) p(\boldsymbol{B}_{t-1}^{(j)} \mid \boldsymbol{X}_t, \boldsymbol{Y}_t) \\
&= p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) \frac{p(\boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t)}{p(\boldsymbol{X}_t, \boldsymbol{Y}_t)} \\
&\propto p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) p(\boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) \\
&\propto p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_{t-1}) p(\boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_{t-1}) p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t). \qquad (18)
\end{aligned}$$

Following [17,21], an efficient trial sampling distribution at time t is $q(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) \triangleq p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t)$. For this trial distribution, the importance weight is updated by

$$\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \frac{p(\boldsymbol{B}_t^{(j)} \mid \boldsymbol{X}_t, \boldsymbol{Y}_t)}{q(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) p(\boldsymbol{B}_{t-1}^{(j)} \mid \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1})} \\
&\propto w_{t-1}^{(j)} \cdot p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}). \qquad (19)
\end{aligned}$$

Note that $p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1})$ can be computed by

$$\begin{aligned}
p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) = \int & p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}, \boldsymbol{\alpha}) \\
& \cdot p(\boldsymbol{\alpha} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) \, \mathrm{d}\boldsymbol{\alpha}. \qquad (20)
\end{aligned}$$

If we assume a Gaussian prior distribution on the coefficients, i.e., $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\alpha}_0, \boldsymbol{\Sigma}_0)$, then we have

$$p(\boldsymbol{\alpha} \mid \boldsymbol{B}_t^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) \propto p(\boldsymbol{Y}_t \mid \boldsymbol{B}_t^{(j)}, \boldsymbol{X}_t, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) \sim \mathcal{N}(\boldsymbol{\alpha}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}), \qquad (21)$$

where

$$\boldsymbol{\Sigma}_t^{(j)} \triangleq \left[\boldsymbol{\Sigma}_0^{-1} + \sum_{i=1}^{t} \boldsymbol{D}_i^{(j)^{\mathrm{T}}} \boldsymbol{D}_i^{(j)} \right]^{-1} \qquad (22)$$

and

$$\boldsymbol{\alpha}_t^{(j)} \triangleq \boldsymbol{\Sigma}_t^{(j)} \left[\boldsymbol{\Sigma}_0^{-1} \boldsymbol{\alpha}_j + \sum_{i=1}^{t} y_i \boldsymbol{D}_i^{(j)^{\mathrm{T}}} \right], \tag{23}$$

where $\boldsymbol{D}_i^{(j)} \triangleq [\boldsymbol{x}_i, \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_{i,1}^{(j)}), \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_{i,2}^{(j)})]$. Define

$$K_t^{(j)} \triangleq \boldsymbol{\Sigma}_{t-1}^{(j)} + \delta_\alpha^2 \boldsymbol{I}, \tag{24}$$

$$\gamma_t^{(j)} \triangleq 1 + \boldsymbol{D}_t^{(j)} K_t^{(j)} \boldsymbol{D}_t^{(j)^{\mathrm{T}}}. \tag{25}$$

Substituting Eq. (21) into Eq. (20), we obtain

$$p(y_t \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_{t-1}, \boldsymbol{Y}_{t-1}) \sim \mathcal{N}(h_t^{(j)}, \gamma_t^{(j)}) \triangleq \rho_t^{(j)} \tag{26}$$

with

$$h_t^{(j)} \triangleq \boldsymbol{D}_t^{(j)} \boldsymbol{\alpha}_{t-1}^{(j)}. \tag{27}$$

Note that the a posteriori mean and covariance of parameters $\boldsymbol{\alpha}_t^{(j)}$ in Eqs. (22) and (23) can be updated recursively as follows. Using the matrix inversion lemma, Eqs. (22) and (23) become

$$\boldsymbol{\alpha}_t^{(j)} = \boldsymbol{\alpha}_{t-1}^{(j)} + \frac{y_t - h_t^{(j)}}{\gamma_t^{(j)}} K_t^{(j)} \boldsymbol{D}_t^{(j)^{\mathrm{T}}}, \tag{28}$$

$$\boldsymbol{\Sigma}_t^{(j)} = \boldsymbol{\Sigma}_{t-1}^{(j)} - \frac{1}{\gamma_t^{(j)}} K_t^{(j)} \boldsymbol{D}_t^{(j)^{\mathrm{T}}} \boldsymbol{D}_t^{(j)} K_t^{(j)}. \tag{29}$$

3.4. Resampling step

The importance sampling weight $w_t^{(j)}$ measures the quality of the corresponding imputed signal sequence $\boldsymbol{\mu}_t^{(j)}$. A relatively small weight implies that the sample is drawn far from the main body of the posterior distribution and has a small contribution in the final estimation. Such a sample is said to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient. To avoid the degeneracy, an useful resampling procedure [17,21] may be used. A resampling method is to multiply the streams with the larger importance weights, while eliminate the ones with small importance weights. A simple resampling procedure consists of the following two steps: (1) Sample a new set of streams $\{\hat{\boldsymbol{\mu}}_t^{(j)}, \hat{\boldsymbol{\alpha}}_t^{(j)}, \hat{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^N$ from $\{\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\alpha}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\}_{j=1}^N$ with probability proportional to the importance weights $\{w_t^{(j)}\}_{j=1}^N$; (2) Assign equal weight to each stream in $\{\hat{\boldsymbol{\mu}}_t^{(j)}, \hat{\boldsymbol{\alpha}}_t^{(j)}, \hat{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^N$, i.e. $\hat{w}_t^j = 1$, $j = 1, \dots, N$.

Resampling can be done at every fixed-length time interval or it can be conducted dynamically. The effective sample size can be used to monitor the variation of the importance weights of the sample streams, and to decide when to resample as the system evolves. The effective sample size is defined as [18]

$$\bar{N}_t \triangleq \frac{N}{1 + v_t^2}, \tag{30}$$

where v_t, the coefficient of variation, is given by $v_t^2 = 1/N\sum_{j=1}^{N}((w_t^{(j)}/\bar{w}_t) - 1)^2$ with $\bar{w}_t = \sum_{j=1}^{N} w_t^{(j)}/N$. In dynamic resampling, a resampling step is performed once the effective sample size $\bar{N}_t$ is below a certain threshold. We use $N = 1000$ in this study. The resampling procedure can be seen as a trade-off between the bias and the variance. That is, the new samples with their weights resulting from the resampling procedure are approximately proper, which introduces small bias in the Monte Carlo estimation. On the other hand, however, resampling significantly reduces the Monte Carlo variance for future samples.

Algorithm summary: Assume that we have properly weighted samples $\{\boldsymbol{\mu}_{t-1}^{(j)}, \boldsymbol{\alpha}_{t-1}^{(j)}, \boldsymbol{\Sigma}_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^{N}$ at time $(t-1)$. At time t, update the samples to obtain $\{\boldsymbol{\mu}_{t}^{(j)}, \boldsymbol{\alpha}_{t}^{(j)}, \boldsymbol{\Sigma}_{t}^{(j)}, w_{t}^{(j)}\}_{j=1}^{N}$ as described above. Finally, we summarize the SMC algorithm for estimating probit regression as follows:

(0) Initialization: for each $j = 1, 2, \ldots, N$: Set $\boldsymbol{\mu}_0^{(j)}$ as the mean of different classes; Set $\boldsymbol{\Sigma}_0^{(j)} = (D^{\mathrm{T}}D)^{-1}$ with $\boldsymbol{D} \triangleq [\boldsymbol{D}_1^{\mathrm{T}}, \ldots, \boldsymbol{D}_n^{\mathrm{T}}]^{\mathrm{T}}$, and draw $\boldsymbol{\alpha}_0^{(j)} \sim \mathcal{N}(0.5\boldsymbol{I}_{1,m+2}, 10\boldsymbol{I}_{m+2}^{(j)})$; Set $w_0^{(j)} = 1$ and $Y_1^{(j)} = 1000y_1$.

Table 1
Strongest genes selected by using bootstrap Bayesian gene selection ($\pi_i = 5/p$) for breast cancer data

Gene no.	Frequency	Index no. (Clone ID)	Gene description
1	0.1416	10 (26184)	Phosphofructokinase, platelet
2	0.1401	336 (823940)	Transducer of ERBB2, 1 (TOP1)
3	0.1372	858 (783729)	v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2
4	0.1234	733 (134748)	Glycine cleavage system protein H (aminomethyl carrier)
5	0.1109	2423 (26082)	Very low density lipoprotein receptor
6	0.1089	955 (950682)	Phosphofructokinase, platelet
7	0.0958	1443 (415698)	Galactosylceramidase (Krabbe disease)
8	0.0837	1417 (825478)	Zinc-finger protein 146
9	0.0801	2428 (26184)	Phosphofructokinase, platelet
10	0.0757	118 (47542)	Small nuclear ribonucleoprotein Dl polypeptide (16 kDa)
11	0.0731	496 (376516)	Cell division cycle 4-like
12	0.0631	1120 (841617)	Human mRNA for ornithine decarboxylase antizyme, ORF 1 and ORF 2
13	0.0530	2018 (139354)	ESTs
14	0.0514	523 (28012)	O-linked N-acetylglucosamine (GlcNAc) transferase
15	0.0498	1766 (239958)	DKFZP586G1822 protein
16	0.0465	2699 (44180)	Alpha-2-macroglobulin
17	0.0420	1859 (307843)	ESTs
18	0.0413	1008 (897781)	Keratin 8
19	0.0370	1179 (788721)	KIAA0090 protein
20	0.0351	555 (548957)	General transcription factor II, i, pseudogene 1

The following steps are implemented at time t ($t = 1, \ldots, n$) to update each weighted sample. For $j = 1, \ldots, N$:

(1) Draw $\boldsymbol{\mu}_t^{(j)}$ according to $p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)}, \boldsymbol{X}_t, \boldsymbol{Y}_t) \propto p(\boldsymbol{\mu}_t^{(j)} \mid \boldsymbol{B}_{t-1}^{(j)})$ in Eq. (12). Append $\boldsymbol{\mu}_t^{(j)}$ to $\boldsymbol{B}_{t-1}^{(j)}$ to obtain $\boldsymbol{B}_t^{(j)}$.

(2) Compute $K_t^{(j)}, \gamma_t^{(j)}, h_t^{(j)}$, and $\rho_t^{(j)}$ given, respectively, by Eqs. (24), (25), (27), and (26).

(3) Compute the importance weight $w_t^{(j)} \propto w_{t-1}^{(j)} \rho_t^{(j)}$.

(4) Update $\boldsymbol{\alpha}_t^{(j)}$ and $\boldsymbol{\Sigma}_t^{(j)}$ according to Eqs. (28) and (29).

(5) Compute the effective sample size $\bar{N}_t$ given by Eq. (30). If $\bar{N}_t \leqslant \lambda N$ (in this study, $\lambda = 0.1$), then perform the following resampling steps to obtain a new set of sample streams:
   - Sample a new set of streams $\{\hat{\boldsymbol{\mu}}_t^{(j)}, \hat{\boldsymbol{\alpha}}_t^{(j)}, \hat{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^N$ from $\{\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\alpha}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\}_{j=1}^N$ with probability proportional to the importance weights $\{w_t^{(j)}\}_{j=1}^N$;
   - Assign equal weight to each stream in $\{\hat{\boldsymbol{\mu}}_t^{(j)}, \hat{\boldsymbol{\alpha}}_t^{(j)}, \hat{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^N$, i.e., $\hat{w}_t^{(j)} = 1/N$, $j = 1, \ldots, N$.

(6) Draw $y_t^{(j)}$ for $j = 1, \ldots, N$ from truncated normal distribution $\mathcal{N}(\boldsymbol{D}_t^{(j)} \boldsymbol{\alpha}_t^{(j)}, 1)$ based on the observation w_t. See the first paragraph of Section 3.2.

Table 2
The sample probabilities for breast cancer data

Sample no.	w	Gibbs sampler		SMC	
		$P(w = 1)$	Classification	$P(w = 1)$	Classification
1	0	0.1688	0	0.0198	0
2	0	0.0984	0	0.0069	0
3	0	0.1308	0	0.0003	0
4	0	0.2557	0	0.0000	0
5	0	0.1644	0	0.0015	0
6	0	0.1258	0	0.0099	0
7	1	0.9738	1	0.9913	1
8	1	0.9899	1	0.9972	1
9	1	0.9830	1	0.9936	1
10	1	0.9794	1	0.9973	1
11	1	0.9654	1	0.8911	1
12	1	0.9760	1	0.9971	1
13	1	0.8022	1	0.9903	1
14	1	0.8985	1	0.9982	1
15	1	0.9815	1	0.9909	1
16	1	0.9984	1	0.9996	1
17	1	0.9077	1	0.8361	1
18	0	0.0055	0	0.0063	0
19	1	0.9995	1	0.9992	1
20	1	1.0000	1	1.0000	1
21	1	0.8422	1	0.9788	1
22	1	0.9871	1	0.9512	1
No. of misclassification			0		0

After obtaining the estimates of all parameters $\boldsymbol{\theta}_n^{(j)} = \{\boldsymbol{\mu}_n^{(j)}, \boldsymbol{\alpha}_n^{(j)}\}$ and $w_n^{(j)}$ for $j = 1, \ldots, N$ using SMC, we then predict the tested sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]$ by

$$P(w = 1|\boldsymbol{x}) = \frac{1}{\sum_{j=1}^{N} w_n^{(j)}} \sum_{j=1}^{N} \Phi(h(\boldsymbol{x}, \boldsymbol{\mu}_n^{(j)}, \boldsymbol{\alpha}_n^{(j)}))w_n^{(j)}, \tag{31}$$

with

$$h(\boldsymbol{x}, \boldsymbol{\mu}_n^{(j)}, \boldsymbol{\alpha}_n^{(j)}) \triangleq \boldsymbol{D}_n^{(j)}\boldsymbol{\alpha}_n^{(j)} = [\boldsymbol{x}, \phi(\boldsymbol{x}, \boldsymbol{\mu}_{n,1}^{(j)}), \boldsymbol{\mu}_{n,2}^{(j)}]\boldsymbol{\alpha}_n^{(j)}.$$

4. Experimental analysis

In the first step of our proposed algorithms, we pre-select genes based on the following criterion: the smaller is the sum of squares within groups and the bigger is the sum of squares between groups, the better is the classification accuracy. Therefore, we can define a score using the above two statistics to pre-select genes, i.e. the ratio of the between-group to within-group sum of squares. Next, the proposed method is tested on several data sets including the hereditary breast cancer data, the small round blue-cell tumor data, and the acute leukemia tumor data.

Table 3
The top 20 important genes selected for breast cancer data for different noise levels ($\pi_i = 5/p$)

Gene no.	$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.5$	
	Frequency	Index no. (Clone ID)	Frequency	Index no. (Clone ID)	Frequency	Index no. (Clone ID)
1	0.1427	858 (783729)	0.1334	336 (823940)	0.1953	2423 (26082)
2	0.1409	1443 (415698)	0.1278	10 (26184)	0.1301	733 (134748)
3	0.1280	10 (26184)	0.1252	858 (783729)	0.1266	336 (823940)
4	0.1175	336 (823940)	0.1199	955 (950682)	0.1164	1443 (415698)
5	0.1172	2699 (44180)	0.0974	2428 (26184)	0.1038	1345 (949932)
6	0.1090	955 (950682)	0.0956	2734 (46019)	0.0993	858 (783729)
7	0.1051	118 (47542)	0.0936	1797 (144926)	0.0762	1531 (711826)
8	0.1007	2428 (26184)	0.0796	2423 (26082)	0.0761	1859 (307843)
9	0.0962	1999 (247818)	0.0794	1179 (788721)	0.0732	10 (26184)
10	0.0520	733 (134748)	0.0769	523 (28012)	0.0698	2342 (284592)
11	0.0397	1859 (307843)	0.0673	1065 (843076)	0.0686	1008 (897781)
12	0.0385	1120 (841617)	0.0637	1008 (897781)	0.0682	2428 (26184)
13	0.0370	2734 (46019)	0.0627	258 (324210)	0.0681	2699 (44180)
14	0.0366	1797 (144926)	0.0595	1859 (307843)	0.0663	1068 (26184)
15	0.0357	809 (810899)	0.0593	496 (376516)	0.0659	585 (293104)
16	0.0352	1008 (897781)	0.0589	1443 (415698)	0.0640	1628 (233365)
17	0.0345	1766 (239958)	0.0579	1345 (949932)	0.0635	2018 (139354)
18	0.0344	2893 (32790)	0.0573	963 (897646)	0.0627	2259 (814270)
19	0.0340	1065 (843076)	0.0572	2761 (47884)	0.0616	1797 (144926)
20	0.0338	1446 (81331)	0.0561	2259 (814270)	0.0614	1466 (767817)

4.1. Breast cancer data

In our first experiment, we will focus on hereditary breast cancer data, which can be downloaded from the web page for the original paper [29]. In [29], cDNA microarrays are used in conjunction with classification algorithms to show the feasibility of using the differences in global gene expression profiles to separate BRCA1 and BRCA2 mutation-positive breast cancers. Twenty-two breast tumor samples from 21 patients were examined: 7 BRCA1, 8 BRCA2, and 7 sporadic. There are 3226 genes for each tumor sample. We use our methods to classify BRCA1 versus the others (BRCA2 and sporadic). The cross-validation (leave-one-out) method is employed to compute all classification errors.

We analyzed the proposed methods based on the natural log ratio data of the breast cancer data. The prior is set as $\pi_i = 5/p$. From Table 1, it is seen gene 10 (phosphofructokinase, platelet) is the most important gene for all methods. It is also a very important gene in [7,10,23]. The second important gene is gene 336 (TOP1), which is another important gene. It interacts with the oncogene receptor ERBB2, and is found to be more highly expressed in BRCA2 and sporadic cancers, which are likewise more likely to harbor ERBB2 gene amplifications [30].

Table 4
The top 20 important genes selected for breast cancer data with different prior setting ($\pi_i = 10/p$ and $\pi_i = 15/p$)

No.	$\pi_i = 10/p$		$\pi_i = 15/p$	
	Frequency	Index no. (Clone ID)	Frequency	Index no. (Clone ID)
1	0.1550	858 (783729)	0.1557	336 (823940)
2	0.1456	336 (823940)	0.1490	858 (783729)
3	0.1084	10 (26184)	0.1338	733 (134748)
4	0.0970	955 (950682)	0.1296	10 (26184)
5	0.0886	2423 (26082)	0.1262	2699 (44180)
6	0.0870	1417 (825478)	0.1239	118 (47542)
7	0.0857	1179 (788721)	0.1208	1443 (415698)
8	0.0855	2428 (26184)	0.1177	1999 (247818)
9	0.0851	733 (134748)	0.1161	1179 (788721)
10	0.0846	523 (28012)	0.1140	1120 (841617)
11	0.0841	1443 (415698)	0.1132	955 (950682)
12	0.0804	2699 (44180)	0.1125	523 (28012)
13	0.0775	2734 (46019)	0.1098	1417 (825478)
14	0.0767	1120 (841617)	0.1086	585 (783729)
15	0.0752	118 (47542)	0.1075	1288 (564803)
16	0.0732	963 (897646)	0.1071	1797 (144926)
17	0.0716	585 (293104)	0.1063	496 (376516)
18	0.0712	1288 (564803)	0.1054	523 (28012)
19	0.0708	1620 (137638)	0.1030	2259 (814270)
20	0.0717	496 (376516)	0.1026	3010 (366824)

Using the top five to ten genes for classification, no error is found. The probability for each sample based on the top five genes using SMC is listed in Table 2. It is seen that the probability for each sample is very close to the true label values (namely 0 and 1). Comparing the Gibbs sampler, the SMC method can get a better estimation of the probabilities in this example.

Sensitivity and robustness: In order to check the sensitivity and robustness of our algorithms, we have added white Gaussian noise with different variances to the data and re-applied our algorithms to the contaminated data. The strongest genes are listed in Table 3. It is seen that gene 10 (phosphofructokinase, platelet) and gene TOB1 (Clone ID 823940) are also very important genes listed in Table 3 for different noise levels. Hence, the proposed methods are robust (not sensitive) to different noise levels. We test the classification results using the top five to ten genes, and no error is found.

To check the sensitivity to the prior distributions, we have rerun our algorithms for different choices $\pi_i = 10/p$ and $\pi_i = 15/p$, respectively. According to Table 4, it is seen the selected genes are almost same as before. Hence the proposed gene selection methods are also not sensitive to the prior setting. Using the top five or ten genes to test the classification result, no error is found.

Table 5
The top 20 important genes selected for SRBCT data ($\pi_i = 10/p$)

Gene no.	Frequency	Index no. (Clone ID)	Gene description
1	0.1168	255 (325182)	Cadherin 2, N-cadherin (neuronal)
2	0.1141	842 (810057)	Cold shock domain protein A
3	0.1047	976 (786084)	Chromobox homolog 1 (Drosophila HP1 beta)
4	0.1024	246 (377461)	Caveolin 1, caveolae protein, 22 kDa
5	0.1023	2050 (295985)	ESTs
6	0.1016	1873 (166195)	Ribonuclease/angiogenin inhibitor
7	0.0997	365 (1434905)	Homeo box B7
8	0.0918	1093 (812965)	v-myc avian myelocytomatosis viral oncogene homolog
9	0.0906	1700 (796475)	ESTs, Moderately similar to skeletal muscle LIM-protein FHL3
10	0.0898	867 (79502)	Methionine adenosyltransferase II, alpha
11	0.0897	1579 (204299)	Replication protein A3 (14 kDa)
12	0.0892	1389 (770394)	Fc fragment of IgG, receptor, transporter, alpha
13	0.0890	1662 (377048)	Homo sapiens incomplete cDNA for a mutated allele of a myosin class I
14	0.0880	153 (383188)	Recoverin
15	0.0879	481 (825411)	*N*-acetylglucosamine receptor 1 (thyroid)
16	0.0861	1601 (629896)	Microtubule-associated protein 1B
17	0.0844	1797 (128126)	Decay accelerating factor for complement
18	0.0841	1347 (743229)	Neurofilament 3 (150 kDa medium)
19	0.0841	742 (812105)	Transmembrane protein
20	0.0828	573 (163174)	Transcription elongation factor A (SII), 1

4.2. Small round blue-cell tumor data

In this experiment, we focus on the small, round blue cell tumors (SRBCTs) of childhood, which include neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS) in [31]. The data set of the four cancers are composed of 2308 genes and 63 samples, where the NB has

Table 6
The sample probabilities for SRBCT data

Sample no.	w	Gibbs sampler		SMC	
		$P(w=1)$	Classification	$P(w=1)$	Classification
1	0	0.0021	0	0.0000	0
2	0	0.0180	0	0.0003	0
3	0	0.1354	0	0.0240	0
4	0	0.0725	0	0.0004	0
5	0	0.0004	0	0.0006	0
6	0	0.0109	0	0.0122	0
7	0	0.0216	0	0.0000	0
8	0	0.0008	0	0.0870	0
9	0	0.0020	0	0.0000	0
10	0	0.0023	0	0.0005	0
11	0	0.0348	0	0.0020	0
12	0	0.0004	0	0.0031	0
13	0	0.0268	0	0.0381	0
14	0	0.0036	0	0.0001	0
15	0	0.0105	0	0.0000	0
16	0	0.0032	0	0.0582	0
17	0	0.0396	0	0.0032	0
18	0	0.0133	0	0.0000	0
19	0	0.0150	0	0.1905	0
20	0	0.2538	0	0.0655	0
21	0	0.0339	0	0.0001	0
22	0	0.0075	0	0.0002	0
23	0	0.0023	0	0.0045	0
24	1	0.9494	1	0.9952	1
25	1	0.9522	1	0.9924	1
26	1	0.9991	1	0.9808	1
27	1	0.9402	1	0.9840	1
28	1	0.9986	1	1.0000	1
29	1	0.8722	1	0.7445	1
30	1	0.6418	1	0.9781	1
31	1	0.9847	1	0.9636	1
32	1	0.9706	1	1.0000	1
33	1	0.9411	1	0.9530	1
34	1	0.9457	1	0.9332	1
35	1	0.9221	1	0.9775	1
No. of misclassification			0		0

12 samples; the RMS has 23 samples; the NHL has eight samples and the EMS has 20 samples. We classify the RMS and NB tumors. The data set for the two cancers is composed of 2308 genes and 35 samples, 23 samples for RMS and 12 samples for NB. The ratio data was truncated from below at 0.01.

Table 5 lists the strongest genes found by the bootstrap Bayesian gene selection with $\pi_i = 10/p$. It is seen that gene 2050 (Clone ID 295985), gene 255 (clone ID 325182), gene 246 (Clone ID 377461), gene 1389 (Clone ID 770394), gene 742 (Clone ID 812105), gene 867 (Clone ID 784593), gene 153 (Clone ID 383188) and gene 1601 (Clone ID 629896) are important genes listed in [23,31]. Using the top five to ten genes for classification, no error is found. The probability for each sample based on the top five genes using SMC is listed in Table 6. It is seen again that the probability for each sample is very close to the true label values.

Table 7
The top 20 important genes selected for acute leukemia data ($\pi_i = 15/p$)

Gene no.	Frequency	Index no. (Clone ID)	Gene description
1	0.1571	6345	GLUL Glutamate-ammonia ligase (glutamine synthase)
2	0.1535	1903	Translational initiation factor 2 beta subunit (elF-2-beta) mRNA
3	0.1407	5402	E2F5 E2F transcription factor 5, pl30-binding
4	0.1386	2056	NFKB1 Nuclear factor of kappa light polypeptide gene enhancer in B-cells 1
5	0.1299	4781	Gp25L2 protein
6	0.1052	1144	SPTAN1 Spectrin, alpha, non-erythrocytic 1 (alpha-fodrin)
7	0.0975	1120	SNRPN Small nuclear ribonucleoprotein polypeptide N
8	0.0835	1551	Importin beta subunit mRNA
9	0.0687	3320	Leukotriene C4 synthase (LTC4S) gene
10	0.0628	6215	MPO from Human myeloperoxidase gene, exons 1–4
11	0.0624	4535	Retinoblastoma binding protein P48
12	0.0563	4142	CD37 CD37 antigen
13	0.0453	4328	Proteasome iota chain
14	0.0451	2242	Peptidyl-prolyl cis–trans isomerase
15	0.0398	2348	ACADM Acyl-Coenzyme A dehydrogenase, C-4 to C-12 straight chain
16	0.0397	3258	Phosphotyrosine independent ligand p62 for the Lck SH2 domain mRNA
17	0.0376	4211	VIL2 Villin 2 (ezrin)
18	0.0365	6539	Epb72 gene exon 1
19	0.0355	6919	RNS2 Ribonuclease 2 (eosinophil-derived neurotoxin; EDN)
20	0.0354	1208	CD44 gene extracted from Human hyaluronate receptor gene

4.3. Acute leukemia data

Following the experimental setup in [1], the data is split into a training set consisting of 38 samples of which 27 are ALL and 11 are AML, and a test set of 34 samples, 20 ALL and 14 AML. In [1], a classifier is trained using a weighted voting scheme on the training samples, and correctly classifies 29 of the 34 samples.

Table 7 lists the 20 strongest genes using bootstrap Bayesian gene selection with $\pi_i = 15/p$. The index number is the Clone ID in this data set. It is seen that genes 6345, 1903, 5402, 2056, and 1144 are the strongest genes. Gene 1144, 1120, and 4535

Table 8
The sample probabilities for acute leukemia data using SMC

Sample no.	w	$P(w=1)$	Classification	Sample no.	w	$P(w=1)$	Classification
1	0	0.0854	0	37	1	0.9390	1
2	0	0.2378	0	38	1	0.9951	1
3	0	0.0032	0	39	0	0.0123	0
4	0	0.0657	0	40	0	0.2192	0
5	0	0.2326	0	41	0	0.2587	0
6	0	0.0417	0	42	0	0.0469	0
7	0	0.0438	0	43	0	0.0255	0
8	0	0.0097	0	44	0	0.0145	0
9	0	0.1649	0	45	0	0.2529	0
10	0	0.0004	0	46	0	0.2328	0
11	0	0.2129	0	47	0	0.1535	0
12	0	0.1720	0	48	0	0.0236	0
13	0	0.1439	0	49	0	0.3763	0
14	0	0.1717	0	50	1	0.8971	1
15	0	0.2821	0	51	1	0.8424	1
16	0	0.0151	0	52	1	0.8054	1
17	0	0.1080	0	53	1	0.7792	1
18	0	0.0311	0	54	1	0.8902	1
19	0	0.0955	0	55	0	0.2577	0
20	0	0.0015	0	56	0	0.2027	0
21	0	0.0000	0	57	1	0.8432	1
22	0	0.1835	0	58	1	0.7189	1
23	0	0.1379	0	59	0	0.0152	0
24	0	0.0701	0	60	1	0.9972	1
25	0	0.0677	0	61	1	1.0000	1
26	0	0.0202	0	62	1	0.8733	1
27	0	0.0064	0	63	1	0.7705	1
28	1	0.8896	1	64	1	0.9628	1
29	1	0.8682	1	65	1	0.7066	1
30	1	0.9927	1	66	1	0.6399	1
31	1	0.9884	1	67	0	0.2899	0
32	1	0.8088	1	68	0	0.1731	0
33	1	0.8726	1	69	0	0.2351	0
34	1	0.8573	1	70	0	0.1501	0
35	1	0.9294	1	71	0	0.0002	0
36	1	0.9230	1	72	0	0.1430	0

are also listed in [7]. Using the top ten genes for classification, no error is found. The probability for each sample using SMC is listed in Table 8.

5. Conclusions

In this paper, we have studied the problem of gene selection and classification based on the expression data. Considering the selection bias problem, we proposed a bootstrap Bayesian gene selection for nonlinear probit regression. A Gibbs sampler is employed to find the strongest genes. Some numerical techniques to speed up the computation is discussed. Bootstrap technique is employed to make the gene selection robust. We then developed a nonlinear probit Bayesian classifier consisting of a linear term plus a nonlinear term, the parameters of which are estimated using the sequential Monte Carlo method. These new methods are applied to analyze several data sets including the hereditary breast cancer data, the small round blue-cell tumor data, and the acute leukemia tumor data. The experimental results show the proposed methods can effectively find important genes, and the classification accuracies are very high.

Note that the authors [32] studied gene selection and classification based on breast cancer data using Pearson coefficients and leave-one-out cross-validation procedure, however it is quite different from our method because we use bootstrap Bayesian method for gene selection and sequential Monte Carlo method for classification.

Acknowledgements

This research was supported by the National Human Genome Research Institute. X. Wang was supported in part by the US National Science Foundation under Grant DMS-0225692.

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, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[2] S. Kim, E.R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J.M. Trent, M. Bittner, Multivariate measurement of gene expression relations, Genomics 67 (2000) 201–209.

[3] I. Tabus, J. Astola, On the use of MDL principle in gene expression prediction, J. Appl. Signal Process. 2001 (4) (2001) 297–303.

[4] X. Zhou, X. Wang, E.D. Dougherty, Construction of genomic networks using mutual-information clustering and reversible-jump Markov-Chain-Monte-Carlo predictor design, Signal Process. 83 (2003) 745–761.

[5] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, L. Lu, D.B. Lewis,

R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, L.M. Staudt, Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403 (2000) 503–511.

[6] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.

[7] K.E. Lee, N. Sha, E.R. Dougherty, M. Vannucci, B.K. Mallick, Gene selection: a Bayesian variable selection approach, Bioinformatics 19 (2003) 90–97.

[8] L. Li, C.R. Weinberg, T.A. Darden, L.G. Pedersen, Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics 17 (2001) 1131–1142.

[9] C.H. Ooi, P. Tan, Genetic algorithms applied to multi-class prediction for the analysis of gene expression data, Bioinformatics 19 (2003) 37–44.

[10] S. Kim, E.R. Dougherty, J. Barrea, Y. Chen, M. Bittner, J.M. Trent, Strong feature sets from small samples, J. Comput. Biol. 9 (2002) 127–146.

[11] H. Chipman, E.I. George, R. McCulloch, The practical implementation of Bayesian model selection, in: P. Lahiri (Ed.), In: Model Selection, IMS Lecture Notes, Vol. 38 (2002) pp. 67–116.

[12] X. Zhou, X. Wang, E.R. Dougherty, Missing value estimation based on linear and non-linear regression with Bayesian gene selection, Bioinformatics 19 (2003) 2302–2307.

[13] R. Jornsten, B. Yu, Simultaneous gene clustering and subset selection for sample classification via MDL, Bioinformatics 19 (2003) 1100–1109.

[14] C. Ambroise, G.J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. USA 99 (2002) 6562–6566.

[15] E.P. Xing, M.I. Jordan, R.M. Karp, Feature selection for high-dimensional genomic microarray data, Proceedings of the 18th International Conference on Machine Learning, Williams College, Williamstown, MA, June 28–July 1, 2001.

[16] J. Albert, S. Chib, Bayesian analysis of binary and polychotomous response data, J. Am. Stat. Assoc. 88 (1993) 669–679.

[17] R. Chen, J.S. Liu, Mixture Kalman filters, J. R. Stat. Soc. B 62 (2000) 493–508.

[18] R. Chen, X. Wang, J.S. Liu, Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering, IEEE Trans. Inf. Theory 46 (2000) 2079–2094.

[19] A. Doucet, N. de Freitas, N.J. Gordon, Sequential Monte Carlo Methods in Practice, Springer, New York, 2001.

[20] A. Kong, J.S. Liu, W.H. Hong, Sequential imputation method and Bayesian missing data problems, J. Am. Stat. Assoc. 89 (1994) 278–288.

[21] J.S. Liu, R. Chen, Sequential Monte Carlo methods for dynamic systems, J. Am. Stat. Assoc. 93 (1998) 1032–1044.

[22] C. Robert, Simulation of truncated normal variables, Stat. Comput. 5 (1995) 121–125.

[23] X. Zhou, X. Wang, E.R. Dougherty, Binarization of microarray data based on a mixture model, J. Mol. Cancer Therapeutics 2 (2003) 679–684.

[24] A.M. Zoubir, B. Boashash, The bootstrap and its application in signal processing, IEEE Signal Process. Mag. 15 (1998) 56–76.

[25] J.R. Gleason, Algorithms for balanced bootstrap simulations, Am. Stat. 42 (1988) 263–266.

[26] W. Pan, C.T. Le, Bootstrap model selection in generalized linear models, J. Agri. Biol. Environ. Stat. 6 (2001) 49–61.

[27] G.A.F, Seber, Multivariate Observations, Wiley, New York, 1984.

[28] N. De Freitas, M. Niranjan, A. Gee, A. Doucet, Sequential Monte Carlo methods to train neural networks models, Neural Comput. 12 (2000) 953–993.

[29] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, B. Wilfond, G. Sauter, O.-P. Kallioniemi, A. Borg, J. Trent, Gene expression profiles in hereditary breast cancer, N. Engl. J. Med. 344 (2001) 539–548.

[30] S. Matsuda, J. Kawamura-Tsuzuku, M. Ohsugi, M. Yoshida, M. Emi, Y. Nakamura, M. Onda, Y. Yoshida, A. Nishiyama, T. Yamamoto, Tob, a novel protein that interacts with p185erbB2, is associated with anti-proliferative activity, Oncogene 12 (1996) 705–713.

[31] J. Khan, J.S. Wei, M. Ringnér, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.

[32] L.J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.