Sparse Bayesian classification and feature selection for biological expression data with high correlations

Yang, Xian; Pan, Wei; Guo, Yike

DOI
10.1371/journal.pone.0189541

Publication date
2017

Document Version
Final published version

Published in
PLoS ONE

RESEARCH ARTICLE

Sparse Bayesian classification and feature selection for biological expression data with high correlations

Xian Yang[1], **Wei Pan**[2], **Yike Guo**[1]*

1 Data Science Institute, Imperial College London, London, SW7 2AZ, United Kingdom, **2** Department of Cognitive Robotics, Delft University of Technology, Delft, Netherlands

* y.guo@imperial.ac.uk

Abstract

Classification models built on biological expression data are increasingly used to predict distinct disease subtypes. Selected features that separate sample groups can be the candidates of biomarkers, helping us to discover biological functions/pathways. However, three challenges are associated with building a robust classification and feature selection model: 1) the number of significant biomarkers is much smaller than that of measured features for which the search will be exhaustive; 2) current biological expression data are big in both sample size and feature size which will worsen the scalability of any search algorithms; and 3) expression profiles of certain features are typically highly correlated which may prevent to distinguish the predominant features. Unfortunately, most of the existing algorithms are partially addressing part of these challenges but not as a whole. In this paper, we propose a unified framework to address the above challenges. The classification and feature selection problem is first formulated as a nonconvex optimisation problem. Then the problem is relaxed and solved iteratively by a sequence of convex optimisation procedures which can be distributed computed and therefore allows the efficient implementation on advanced infrastructures. To illustrate the competence of our method over others, we first analyse a randomly generated simulation dataset under various conditions. We then analyse a real gene expression dataset on embryonal tumour. Further downstream analysis, such as functional annotation and pathway analysis, are performed on the selected features which elucidate several biological findings.

OPEN ACCESS

Citation: Yang X, Pan W, Guo Y (2017) Sparse Bayesian classification and feature selection for biological expression data with high correlations. PLoS ONE 12(12): e0189541. https://doi.org/10.1371/journal.pone.0189541

Editor: Bin Liu, Harbin Institute of Technology Shenzhen Graduate School, CHINA

Received: June 21, 2017

Accepted: November 27, 2017

Published: December 27, 2017

Data Availability Statement: All relevant data are available from the figshare repository at the following URL: https://doi.org/10.6084/m9.figshare.5678806.v1.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Biological system is being comprehensively profiled by various expression data through high-throughput technologies, such as gene expression data (measured by the microarray or next generation sequencing technology), protein expression expression (measured by the mass spectrometry-based flow cytometer) and medical imaging (measured by functional magnetic resonance imaging or computerised tomography scan) [1, 2]. Computational and statistical methods for discovering functional roles of features from expression data are required to have the ability of handling large scale datasets. A straightforward analysis is to carry out statistical tests to identify differentially expressed features between groups of samples [3]. Functional analyses, such as the Gene Set Enrichment Analysis (GSEA) [4], can be followed to discover pathways or biological functions that are over-expressed in the differential feature list. Then the biological semantics of differential features can be explored. Besides differential feature discovery, another important type of analysis is sample classification, in which case patient samples are classified by characteristics such as disease subtypes and treatment strategies [5]. The classification model constructed from biological expression data can be used for disease diagnosis [6, 7] or clinical outcome prediction [8, 9]. In this paper, we focus on investigating classification methods to build predictive models from biological expression data.

There are a large range of machine learning methods to construct classification models. Examples of such methods include deep learning [10, 11], graphical models [12, 13], regularised Bayesian models [14], nonparametric Bayesian models [15, 16] and also some traditional methods such as support vector machine (SVM) [17], linear discriminant analysis [18] and Gaussian Naive Bayes [19]. Many tools are particularly designed for biological data. For example, a Python package called Pse-Analysis [20], is developed to automatically generate classifiers for genomics and proteomics datasets. It is based on the framework of LIBSVM [21] and inherits the characteristics of the SVM method. Pse-Analysis carries out the following five key tasks: feature extraction, parameter selection, model training, cross validation and evaluation. Another example is the iDHS-EL method mainly designed for DNA sequence data [22]. iDHS-EL uses three different ways to extract feature vector to represent sequence data, which leads to three different basic random forest (RF) predictors. Then the ensemble predictor is formed by using these three predictors. RF is a combination of decision trees, which has the ability to select important features.

Among various classification methods, sparse Bayesian Learning (SBL) [23, 24] is featured in overcoming the dimensionality problem. This is a common issue of applying classification methods to large scale data: the number of features is usually several orders of magnitudes over the samples. In our application, feature size (number of measured features) is much larger than sample size. In this case, many classification methods cannot work as well as SBL. For example, authors in [25, 26] find that SBL outperforms decision tree methods. SBL only uses a small subset of input features for prediction, based on the observation that relevant features are sparse compared to the dimension of whole feature space. Bayesian inference is adopted to obtain solutions for probabilistic classification. SBL is in the same functional form of SVM, but provides probabilistic classification. Unlike SVM, benefited from Bayesian formulation, SBL avoids the setting of free parameters which requires the cross-validation process. More specifically, SBL uses a fully probabilistic framework and introduces a prior over the model weights governed by a set of hyperparameters. Each hyperparameter is associated with each weight, whose most probable value is iteratively estimated from the data through an expectation-maximisation (EM) procedure. Sparsity is achieved by finding posterior distributions of many weights are sharply peaked at zero.

One of the distinguishing characteristics of SBL is that it does not only build a classification model but also returns a set of features with non-zero weights. In our application, these features are predictive molecules, differentiating two groups of samples. Thus, we can also regard the SBL as a feature selection method. Traditional feature selection methods include statistics tests to reduce feature space by examining whether the significant values of features of a test pass the predefined threshold. For biological data, there are many advanced feature selection methods being proposed. For example, the binary particle swarm optimisation (BPSO) based model is proposed in [27] for the gene selection of Microarray data. To improve the performance of feature selection, BPSO uses gene-to-class sensitivity (GCS) information in the feature selection process. GCS information is obtained from gene expression data indicating whether a gene is sensitive to sample classes. To evaluate candidate gene subsets selected from BPSO, extreme learning machine (ELM) is used for classification model construction. Unlike this method, SBL, similar to other embedded feature selection methods, integrates the feature selection step into the predictive model construction.

There are some examples of embedded feature selection methods which achieve the feature selection by imposing regularisation on existing classification methods, such as regularised SVM [28, 29] and sparse logistic regression [30, 31]. These methods need to tune the regularization parameters via the cross validation process. The work in [32] develops a Bayesian approach based on a probit regression model with a generalised singular *g*-prior distribution for regression coefficients. The hyperparameters need to be predefined in the model and the selected feature set is quite small. With limited number of selected features, it is not easy to discover biological functions. The main reason that only a quite small feature set is selected is correlated features are not selected coherently. As we know that in a biological process, multiple molecules are working together, resulting in correlated feature expression levels. SBL, imposed with sparsity constraints, cannot simultaneously select correlated features. Instead, one out of a set of correlated features is usually selected in the predictive model. In the literature, several approaches have been proposed for classification and feature selection and some of them are actually based on SBL [33–35]. Not surprisingly, the feature set generated by the Bayesian selection method in [35] is quite small that only 12 out of 7128 genes in an example gene expression dataset are selected. The feature set returned from these methods cannot be easily used to discover predictive pathways or biological functions.

Motivated by the fact that existing classification methods either fail to identify a list of predictive features or easily discard correlated features, we propose a computational method derived from SBL to simultaneously build a classifier and select predictive features which are highly correlated. Our classification model is constructed through an iterative convex optimisation procedure instead of a one-step closed form calculation. Moreover, the optimisation-centric formulation of our method can be easily paralleled [36]. The cost function is cast using hierarchical Bayesian model, where the parameters' prior distribution is parameterised by the hyperparameter. Its main goal is to infer the posterior distribution of parameter via Bayes' rule. Rather than using the EM procedure to update parameters and hyperparameters, our method infer both parameters and hyperparameters via the convex optimisation procedure [24, 37]. The paper is organised as follows, we first detailed the method and then test against simulated and real datasets. In the simulation study, we compare the performance of our method with other methods in the aspects of classification and feature selection abilities. In the real dataset analysis, we apply our method to construct a classification model to predict different types of tumours. The selected features in the predictive model are fed into downstream analyses for biological functions and pathways discovery. The results show that our method can perform classification and feature selection at the same time, while the selected features can give insight into new functional modules.

Methods

The proposed classification method constructs a mathematical model through an iterative optimisation procedure. The resulting model can simultaneously perform classification and return a relevant feature set. In this section, we first follow the sparse Bayesian approach to define a single target optimisation function to obtain both parameters and hyperparameters. Then we infer the equations to iteratively solve the optimisation problem via the smooth-concave procedure. The standard EM updating procedure is replaced by the optimisation procedure.

Optimisation problem definition

Suppose we get a set of input vectors $\{\mathbf{x}_n\}_{n=1}^{N}$ along with corresponding targets $\{y_n\}_{n=1}^{N}$. We wish to learn the underlying functional mapping which is defined by a parameterised function $f(\mathbf{x};\boldsymbol{\beta}) = \sum_{i=1}^{M} \beta_i \phi_i(\mathbf{x})$, where the output is the linear weighted sum of M basis functions and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_M]^\top$ contains the parameter. Let Φ be the $N \times M$ design matrix with $\Phi = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]^\top$, wherein $\boldsymbol{\phi}(\mathbf{x}_n) = [\phi_1(\mathbf{x}_n), \phi_2(\mathbf{x}_n), \ldots, \phi_M(\mathbf{x}_n)]$. Then we can express the mapping function as $f(\mathbf{x};\boldsymbol{\beta}) = \Phi\boldsymbol{\beta}$. Usually, the prior distribution of the weights is assumed to follow a zero-mean isotropic Gaussian:

$$\mathcal{P}(\boldsymbol{\beta}|\boldsymbol{\gamma}) = \prod_{i=1}^{M} \mathcal{N}(\beta_i|0,\gamma_i) = \prod_{i=1}^{M} (2\pi\gamma_i)^{-\frac{1}{2}} \exp\{-\frac{\beta_i^2}{2\gamma_i}\}, \tag{1}$$

where

$$\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_M] \in \mathbb{R}^M, \ \ \boldsymbol{\Gamma} = \mathrm{diag}[\boldsymbol{\gamma}]. \tag{2}$$

Here, we focus on investigating the case that the target variable is binary. The likelihood function $\mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})$ is expressed in the form of the logistic regression model:

$$\log \mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x}) = -\sum_{n=1}^{N} \log[1 + \exp\{-y_n \boldsymbol{\beta}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}]. \tag{3}$$

According to the Bayes' rule, the posterior distribution over weights $\mathcal{P}(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\gamma})$ is proportional to $\mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\mathcal{P}(\boldsymbol{\beta}|\boldsymbol{\gamma})$. Maximisation of posterior is equivalent to finding the maximum over $\boldsymbol{\beta}$ of

$$\log \mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\mathcal{P}(\boldsymbol{\beta}|\gamma) = -\sum_{n=1}^{N} \log[1 + \exp\{-y_n \boldsymbol{\beta}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}] - \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{\Gamma}^{-1}\boldsymbol{\beta} - \frac{1}{2}\log|\boldsymbol{\Gamma}|. \tag{4}$$

Thus, the weights can be found through

$$\underset{\boldsymbol{\beta}}{\mathrm{argmin}}\, E(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \sum_{n=1}^{N} \log[1 + \exp\{-y_n \boldsymbol{\beta}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}] + \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{\Gamma}^{-1}\boldsymbol{\beta} + \frac{1}{2}\log|\boldsymbol{\Gamma}|. \tag{5}$$

The gradient and Hessian matrices at arbitrary point of $\boldsymbol{\beta}^*$ are defined as

$$\mathbf{g}(\boldsymbol{\beta}^*) \triangleq \nabla E(\boldsymbol{\beta})|_{\boldsymbol{\beta}^*} = -\sum_{n=1}^{N} y_n \boldsymbol{\phi}(\mathbf{x}_n)[1 - S\{y_n \boldsymbol{\beta}^{*\top} \boldsymbol{\phi}(\mathbf{x}_n)\}] + \boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}^* \tag{6}$$

and

$$\mathbf{H}(\boldsymbol{\beta}^*) \triangleq \nabla\nabla E(\boldsymbol{\beta})|_{\boldsymbol{\beta}^*} = \boldsymbol{\Phi}^\top \mathrm{diag}\,[\mathbf{y}]\mathbf{Z}(\boldsymbol{\beta}^*)\mathrm{diag}\,[\mathbf{y}]\boldsymbol{\Phi} + \boldsymbol{\Gamma}^{-1} \tag{7}$$

where

$$\mathbf{Z}(\boldsymbol{\beta}^*) = \mathrm{diag}(\{S\{y_n \boldsymbol{\beta}^{*\top} \boldsymbol{\phi}(\mathbf{x}_n)\}[1 - S\{y_n \boldsymbol{\beta}^{*\top} \boldsymbol{\phi}(\mathbf{x}_n)]\}_{n=1}^{N}) \tag{8}$$

and

$$S\{f\} = \frac{1}{(1 + e^{-f})}. \tag{9}$$

The hyperparameter $\boldsymbol{\gamma}$ is updated by maximising the marginal likelihood, which is equivalent to

$$\underset{\gamma}{\operatorname{argmin}} -\mathcal{P}(\mathbf{y}|\boldsymbol{\gamma},\mathbf{x}) = \underset{\gamma}{\operatorname{argmin}} -\int \mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\mathcal{P}(\boldsymbol{\beta}|\gamma)d\boldsymbol{\beta}. \tag{10}$$

According to Taylor expansion, we can get the following approximation at mode $\boldsymbol{\beta}^*$:

$$-\log \mathcal{P}(\mathbf{y}|\boldsymbol{\beta},\mathbf{x})\mathcal{P}(\boldsymbol{\beta}|\gamma) = E(\boldsymbol{\beta}^*) + \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^\top \mathbf{H}(\boldsymbol{\beta}^*)(\boldsymbol{\beta}-\boldsymbol{\beta}^*). \tag{11}$$

Therefore, the logarithm of negative marginal likelihood is

$$-\log \mathcal{P}(\mathbf{y}|\gamma,\mathbf{x}) = E(\boldsymbol{\beta}^*) + \frac{M}{2}\log 2\pi + \frac{1}{2}\log|\mathbf{H}(\boldsymbol{\beta}^*)|. \tag{12}$$

Thus $\boldsymbol{\gamma}$ can be estimated by

$$\underset{\gamma}{\operatorname{argmin}}\, E(\boldsymbol{\beta}^*) + \frac{1}{2}\log|\mathbf{H}(\boldsymbol{\beta}^*)|. \tag{13}$$

From Eqs (5) and (13), we can jointly estimate $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ through a common objective function:

$$\underset{\boldsymbol{\beta},\gamma}{\operatorname{argmin}} \sum_{n=1}^{N} \log[1+\exp\{-y_n\boldsymbol{\beta}^\top\boldsymbol{\phi}(\mathbf{x}_n)\}] + \boldsymbol{\beta}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta} + \log|\boldsymbol{\Gamma}| + \frac{1}{2}\log|\mathbf{H}(\boldsymbol{\beta}^*)|, \tag{14}$$

where $|\mathbf{H}(\boldsymbol{\beta}^*)|$ is the Hessian matrix calculated at mode $\boldsymbol{\beta}^*$, which is assumed to be obtained through the minimisation step of $\boldsymbol{\beta}$ in the iterative optimisation process. The reason we label * in $|\mathbf{H}(\boldsymbol{\beta}^*)|$ is to emphasize that the term $|\mathbf{H}|$ is not involved when updating $\boldsymbol{\beta}$ according to Eq (5).

Iterative optimisation algorithm

The objective function defined in Eq (14) can be formulated as a convex-concave procedure (CCCP) for updating $\boldsymbol{\beta}$ and γ, which is

$$\underset{\boldsymbol{\beta},\gamma}{\operatorname{argmin}}\, u(\boldsymbol{\beta},\gamma) + v(\gamma), \tag{15}$$

where the data fitting term $u(\boldsymbol{\beta}, \gamma)$ is a smooth function in the form of

$$u(\boldsymbol{\beta},\gamma) \triangleq \sum_{n=1}^{N} \log[1+\exp\{-y_n\boldsymbol{\beta}^\top\boldsymbol{\phi}(\mathbf{x}_n)\}] + \boldsymbol{\beta}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}, \tag{16}$$

and the regularisation term $v(\boldsymbol{\gamma})$ is a concave function [24, 38]:

$$v(\gamma) = \log|\boldsymbol{\Gamma}| + \log|\mathbf{H}(\boldsymbol{\beta}^*)|. \tag{17}$$

By expressing the objective function in the convex-concave form [39], we can evoke standard iterative optimisation procedure to get its solution at the $k + 1^{\text{th}}$ iteration as follows:

$$\boldsymbol{\beta}^{k+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, u(\boldsymbol{\beta},\gamma^k) \tag{18}$$

and

$$\gamma^{k+1} = \underset{\gamma}{\operatorname{argmin}}\, u(\boldsymbol{\beta}^{k+1},\gamma) + \nabla_\gamma v(\gamma,\mathbf{H}(\boldsymbol{\beta}^{k+1}))\big|_{\gamma=\gamma^k}^\top \cdot \gamma. \tag{19}$$

If we define

$$w_i^k \triangleq \frac{1}{\sqrt{\gamma_i^k}}, \tag{20}$$

$\boldsymbol{\beta}^{k+1}$ can be obtained as the following expression in the form of reweighted ℓ_2 regularisation:

$$\boldsymbol{\beta}^{k+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{n=1}^{N} \log[1 + \exp\{-y_n \boldsymbol{\beta}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}] + \sum_{i=1}^{M} \| w_i^k \cdot \boldsymbol{\beta} \|_{\ell_2}. \tag{21}$$

Let us use the following notation:

$$\begin{aligned} \boldsymbol{\alpha}^{k+1} &\triangleq \nabla_\gamma v\left(\boldsymbol{\gamma}, \mathbf{H}(\boldsymbol{\beta}^{k+1})\right)^\top |_{\gamma=\gamma^k} \\ &= \nabla_\gamma \left(\log|\Gamma| + \log|\mathbf{H}(\boldsymbol{\beta}^{k+1})|\right)^\top |_{\gamma=\gamma^k} \\ &= [\alpha_1^{k+1}, \quad \cdots \quad , \alpha_N^{k+1}]. \end{aligned} \tag{22}$$

According to the matrix derivative rule, we can derive the expression of the ith element of $\boldsymbol{\alpha}^k$ as:

$$\alpha_i^{k+1} = -\frac{\mathbf{H}(\boldsymbol{\beta}^{k+1})_{i,i}^{-1}}{(\gamma_i^k)^2} + \frac{1}{\gamma_i^k}. \tag{23}$$

According to Eq (19), optimal γ_i^{k+1} is obtained by minimising

$$\frac{(\beta_i^{k+1})^2}{\gamma_i} + \alpha_i^{k+1}\gamma_i. \tag{24}$$

Since

$$\frac{(\beta_i^{k+1})^2}{\gamma_i} + \alpha_i^{k+1}\gamma_i \geq 2\left|\sqrt{\alpha_i^{k+1}} \cdot \beta_i^{k+1}\right|, \tag{25}$$

the optimal $\boldsymbol{\gamma}$ can be obtained as:

$$\gamma_i^{k+1} = \frac{|\beta_i^{k+1}|}{\sqrt{\alpha_i^{k+1}}}, \forall i. \tag{26}$$

The pseudo code is summarised in Algorithm 1.

Algorithm 1 Reweighted ℓ_2 type algorithm

```
Initialise the unknown hyperparameter $\boldsymbol{\gamma}^1$ as a unit vector;
Initialise $w_i^1 = 1, \forall i$;
for $k = 1, \ldots, k_{\max}$ do
    $\boldsymbol{\beta}^{k+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{n=1}^{N} \log[1 + \exp\{-y_n \boldsymbol{\beta}^\top \boldsymbol{\phi}(\mathbf{x}_n)\}] + \sum_{i=1}^{M} \| w_i^k \cdot \beta_i \|_{\ell_2};$   (27)
    $\alpha_i^{k+1} = -\frac{\mathbf{H}(\boldsymbol{\beta}^{k+1})_{i,i}^{-1}}{(\gamma_i^k)^2} + \frac{1}{\gamma_i^k}$;
    $\gamma_i^{k+1} = \frac{|\beta_i^{k+1}|}{\sqrt{\alpha_i^{k+1}}}$;
    $w_i^{k+1} = \frac{1}{\sqrt{\gamma_i^{k+1}}}$;
    if a stopping criterion is satisfied then
        Break.
    end if
end for
```

The cost function defined in Eq (27) of Algorithm 1 is convex, making it possible to apply standard solvers to obtain a global optimal solution. For example, we can consider iterative solvers, such as the standard gradient method, the Newton method and its variants. We should note that in the 5th line of Algorithm 1, the inverse of Hessian matrix needs to be calculated. When the number of features is quite large, which is usually the case in the gene expression data, it is quite computational expensive to do the Hessian matrix inversion. Therefore, we apply a Quasi-Newton method, limited memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm, to directly generate the inverse of Hessian matrix in the iterative solver without performing matrix inversion [40].

Results

In this section, a set of simulated datasets are first used to show the ability of selecting relevant features as well as constructing predictive models. Our method is compared with the other two representative methods: ℓ_1 regularised logistic regression implemented by alternating direction method of multipliers (ADMM) [41] (code is downloaded from https://web.stanford.edu/$\sim$boyd/papers/admm/logreg-l1/) and SBL (code is downloaded from http://www.relevancevector.com). The results will highlight the performance of our method on datasets with high correlations. To demonstrate the applicability of our method on real datasets, we choose a publicly available gene expression dataset for illustration. The identified features (genes) are fed into downstream analyses. We find that the detected functional terms agree with the findings from literature. The analyses of real data show that our method can generate a list of predictive genes that are both used for classifier construction and biological functionality discovery.

Simulated data analysis

In the simulation, we first generate a dataset with the sample and feature size of 500 and 50. We assume there are only 8 non-zero elements in $\boldsymbol{\beta}$. To check the ability of finding correlated features, we split the features with non-zero weights into two sets, S^1 and S^2, each of which contains 4 true features. Then, we initialise a design matrix $\Phi = \mathbf{x}$ with 500 samples and 50 features from the normal distribution. Let $\phi_i(\mathbf{x})$ be the ith column of Φ. We set $\boldsymbol{\phi}_{S^2_m}(\mathbf{x}) = \boldsymbol{\phi}_{S^1_m}(\mathbf{x}) + \mathcal{N}(0, 0.1)$, where S^1_m and S^2_m denote the mth element in each feature set. In this way, we get data generated from correlated features. The target variable vector $\{y_n\}_{n=1}^{500}$ is generated by the linear model using $\boldsymbol{\beta}$ and Φ with additive independent identical distributed Gaussian noises, where the standard deviation of noise varies ranging from {0, 0.1, 0.5, 1}.

We first work on the whole dataset of 500 samples with no additive noise to investigate the ability of selecting the true features. The estimated weights from our method is compared with the true weights, the estimates from the ℓ_1 regularised logistic regression method (RLR) [41] and the sparse Bayesian learning method. We scale the estimated results to make the first real nonzero feature have the same value. The results are shown in Fig 1. It shows that only our method can successfully detect all the true features with magnitudes of the corresponding weights significantly larger than the others. RLR and SBL easily ignore some features that one out of a correlated pair can be detected. This observation is quite encouraging, showing that our method can be potentially used as a good tool to select relevant features. Most classification methods are not guaranteed to have this characteristic.

We then investigate the performance of our method with different levels of noise. The performance is evaluated by the 10-fold cross validation. In the cross validation process, different training dataset is used for feature selection and predictive model construction in each fold. Therefore, selected feature sets from all folds may vary due to the variation of training datasets.

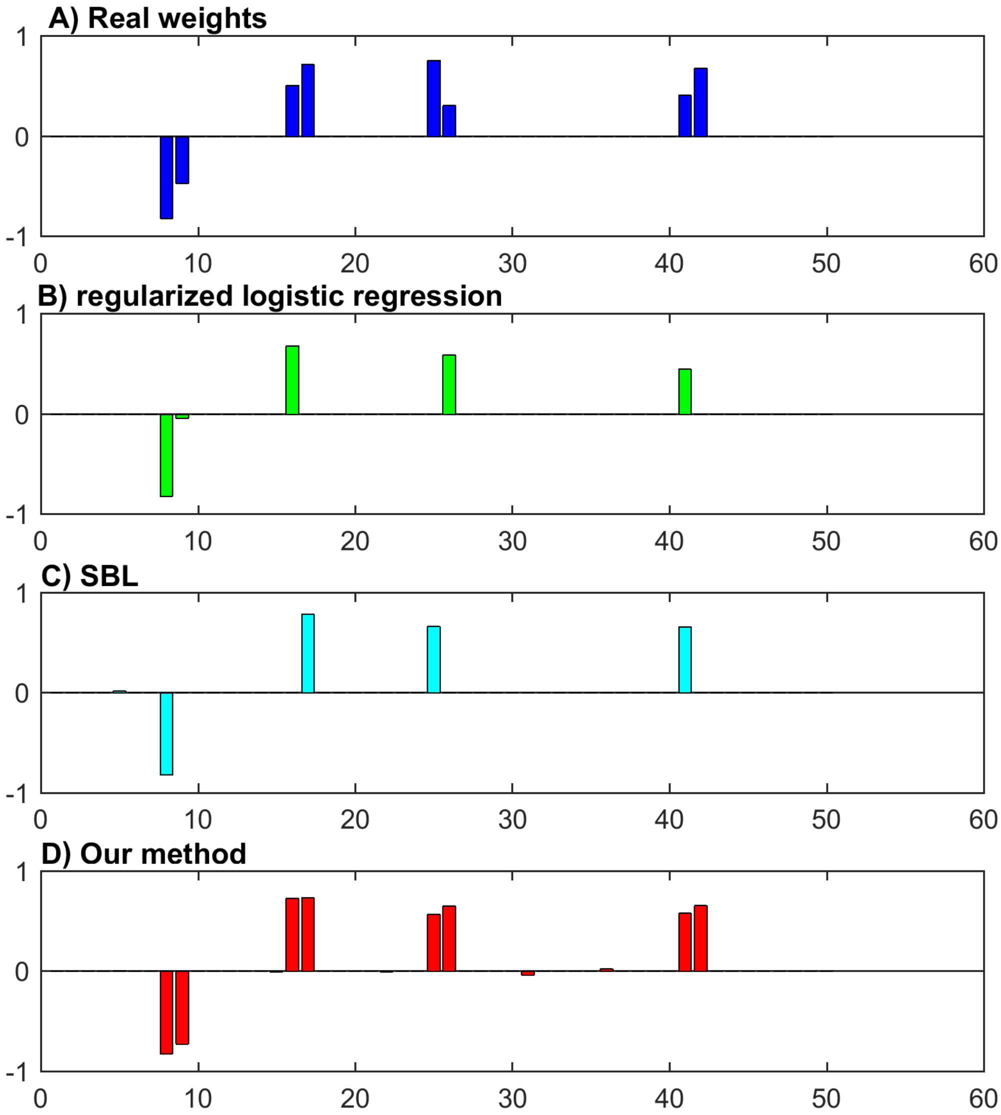

Fig 1. The comparison of the real weights and the weights estimated from different methods.

https://doi.org/10.1371/journal.pone.0189541.g001

To select a feature set which is stable with small fluctuations of the input dataset and also has good predictive accuracy, we use the method from [42]. In the *k*th fold of cross validation, the whole dataset $\mathbf{D}$ is split into two subsets: CV training dataset $\mathbf{D_k}$ and CV testing dataset $\mathbf{D_{\backslash k}}$. Our method can work as a feature selection method on the training dataset $\mathbf{D_k}$ to rank and select the top q features, labelled as $\mathbf{V_{q,k}}$. After features have been selected, our method then constructs a predictive model for classification using $\mathbf{V_{q,k}}$. The prediction results at this CV fold are recorded for later evaluation. To get the complete prediction results, we repeat the above steps for all folds of CV. The method presented in [42] returns an optimal feature set $\tilde{\mathbf{V}}_\mathbf{q}$ with an associated performance score $\tilde{P}_q$ under each value of q. The score $\tilde{P}_q$ is calculated according to the 6th strategy proposed in [43] to assess the prediction accuracy and stability of features (the details of calculating this score can be found in [43]). By checking the maximum value of $\tilde{P}_q$, we can determine the optimal value of q and the corresponding optimal

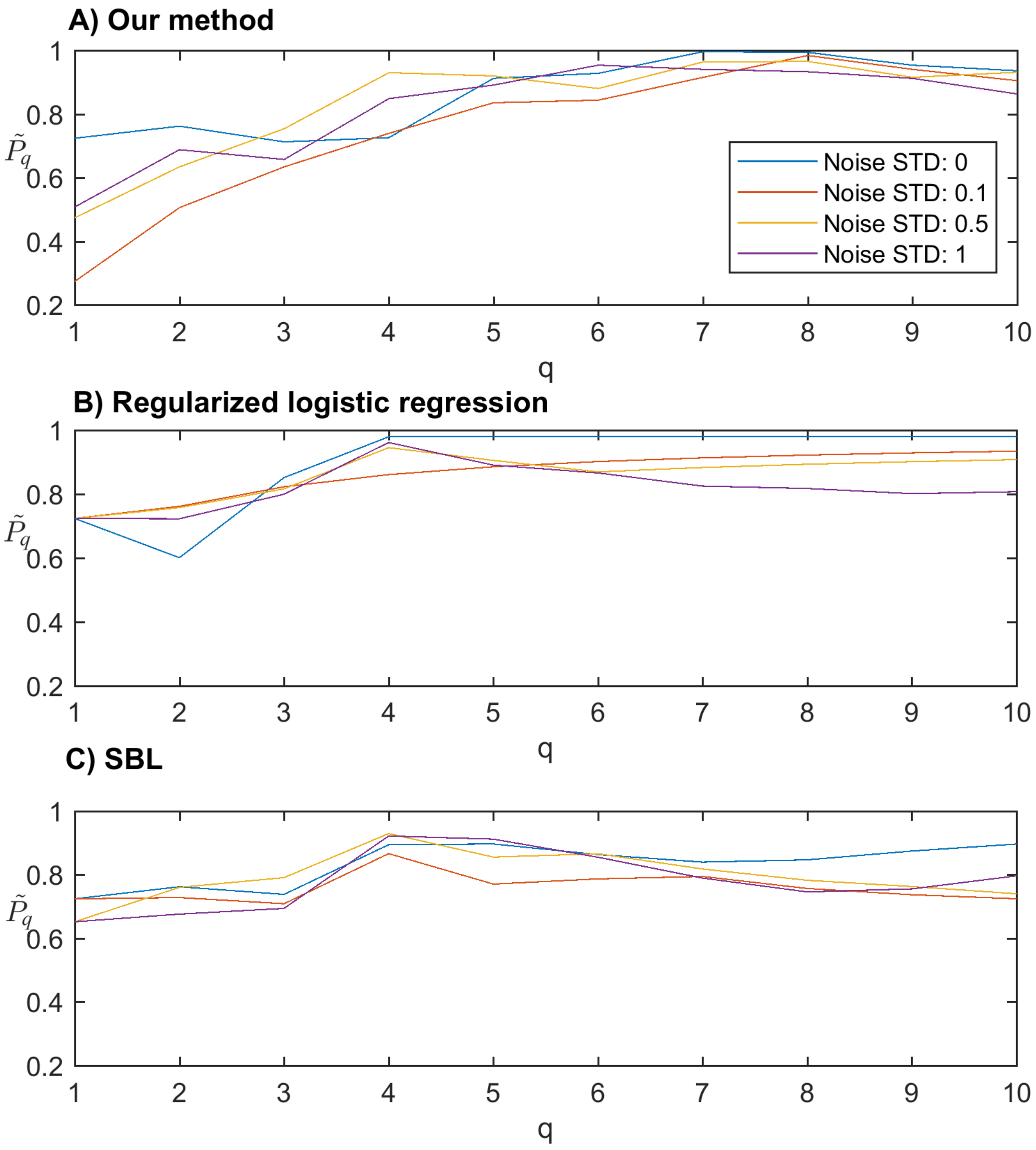

Fig 2. The scores achieved by different methods with the size of feature set varying from 1 to 10 and standard deviation of noise chosen from 0, 0.1, 0.5 and 1.

https://doi.org/10.1371/journal.pone.0189541.g002

feature set $\tilde{\mathbf{V}}_{\mathbf{q}}$. The detected optimal set can then be used to construct a predictive model for future prediction.

In Fig 2, it shows the change of scores $\tilde{P}_q$ under different settings of feature size q and noise level. From Fig 2, we can see that the scores of our method under different noise levels peak when the value of q is close to the number of real nonzero features. In contrast to our method, the optimal value of q for the RLR method and the SBL method are around 4, which is different from the real number of nonzero features. From Fig 2, we can expect that our method works better than the RLR method with respect to feature selection. Table 1 further confirms our observations by showing the accuracy values of classification through the 10-fold cross validation. The feature set size q under different conditions is chosen to be the optimal value detected from Fig 2. To show the ability of detecting real positives (i.e. real nonzero features),

Table 1. The results of our method, the regularised logistic regression (RLR) and sparse Bayesian learning (SBL) under different noise levels. The accuracy of classification, false positive and false negative rates are compared.

Noise STD	Method	Accuracy	False Positive	False Negative
0	Our method	0.996	0	0
	RLR	0.982	0	0.5
	SBL	0.998	0.024	0.5
0.1	Our method	0.986	0	0
	RLR	0.986	0.143	0.5
	SBL	0.992	0	0.5
0.5	Our method	0.98	0	0
	RLR	0.972	0	0.5
	SBL	0.98	0	0.5
1	Our method	0.972	0.024	0.125
	RLR	0.964	0	0.5
	SBL	0.972	0	0.5

https://doi.org/10.1371/journal.pone.0189541.t001

we also present the results of false positive (FP) and false negative (FN) rates in Table 1. We can see that under different noise levels, the accuracy achieved by different methods are all maintained at high levels. However, the false negative rates from other methods are much higher than the rates from our methods. This is because, the other methods cannot detect correlated features that some real features are ignored.

The above observations are only for datasets with fixed sample and feature sizes (500 and 50 respectively). To test generality, we generate a set of datasets as follows: the sample and feature sizes are chosen from {50, 100, 500}; the ratio of non-zero features (sparsity) is either 0.1 or 0.2; non-zero features can be fully independent or 50% of them are highly correlated. For each combination of settings, 20 randomly sampled datasets are generated. We apply our method and also the other two methods on these datasets. The FN, FP and accuracy values are derived from the average values of 20 randomly sampled datasets under each specific setting. The results for the datasets with the sparsity of 10% are recorded in Table 2. The results for the datasets with the sparsity of 20% can be found in S1 Table.

We can see that for the datasets with independent features, the performance of our method is similar with the other two methods. However, for the datasets with correlated features, our method works much better than the other two methods with lower FN and FP rates and similar accuracy values. Especially, the FN rates are much smaller, indicating that the real non-zero features are more likely to be detected by our method. This characteristic of our method is quite important, since biological expression data contains a lot of correlated features. We can also observe a general trend that: with a fixed sample size, the performance of all three methods is reduced when the feature size increases; with a fixed feature size, the performance of all three methods can be improved with more samples. When the proportion of non-zero feature increases to 0.2 (see S1 Table), the performance of all methods are deteriorated. This is because, all these methods are based on the assumption that the feature space is sparse. For biological expression data, the ratio of predictive features or biomarkers is quite low, which is much smaller than 0.2 or even 0.1.

Embryonal tumour gene expression data analysis

We use a public available gene expression dataset of the central nervous system embryonal tumours from the study in [8]. All relevant data are available from the figshare repository at

Table 2. The performance of our method, the RLR method and the SBL method with the sparsity of 0.1.

%cor	# Sample	# Feature	Method	Accuracy	False Positive	False Negative
0	50	50	Our	0.84	0.0156	0.17
			RLR	0.862	0.0189	0.17
			SBL	0.874	0.0211	0.12
		100	Our	0.686	0.111	0.395
			RLR	0.704	0.108	0.425
			SBL	0.656	0.179	0.61
		500	Our	0.534	0.158	0.746
			RLR	0.52	0.211	0.896
			SBL	0.498	0.219	0.974
	100	50	Our	0.972	0	0
			RLR	0.953	0	0
			SBL	0.97	0	0
		100	Our	0.809	0.0267	0.14
			RLR	0.812	0.025	0.125
			SBL	0.859	0.0233	0.035
		500	Our	0.596	0.139	0.627
			RLR	0.586	0.1941	0.747
			SBL	0.567	0.21	0.892
	500	50	Our	0.993	0	0
			RLR	0.983	0	0
			SBL	0.995	0	0
		100	Our	0.988	0	0
			RLR	0.971	0	0
			SBL	0.986	0	0
		500	Our	0.791	0.0357	0.121
			RLR	0.803	0.021	0.139
			SBL	0.811	0.0666	0.049
50	50	50	Our	0.952	0.0022	0.04
			RLR	0.934	0.0189	0.42
			SBL	0.962	0.0844	0.39
		100	Our	0.811	0.0578	0.12
			RLR	0.794	0.0794	0.515
			SBL	0.809	0.1689	0.52
		500	Our	0.549	0.148	0.633
			RLR	0.545	0.214	0.927
			SBL	0.556	0.219	0.974
	100	50	Our	0.985	0	0
			RLR	0.964	0.0011	0.39
			SBL	0.987	0.0622	0.4
		100	Our	0.959	0	0
			RLR	0.952	0.0017	0.49
			SBL	0.969	0.0894	0.48
		500	Our	0.634	0.139	0.45
			RLR	0.631	0.199	0.796
			SBL	0.592	0.2113	0.902
	500	50	Our	0.995	0	0
			RLR	0.982	0.106	0.37
			SBL	0.995	0.0378	0.39
		100	Our	0.995	0	0
			RLR	0.98	0.0628	0.465
			SBL	0.992	0.131	0.48
		500	Our	0.938	0.0026	0.023
			RLR	0.931	0.00011	0.501
			SBL	0.592	0.145	0.455

https://doi.org/10.1371/journal.pone.0189541.t002

the following URL: https://doi.org/10.6084/m9.figshare.5678806.v1. We selected 10 CNS medulloblastomas (MD) samples and 10 non-neuronal origin malignant gliomas (Mglio) samples to show the performance of our method in classifying two tumour types. The samples were hybridised on Affymetrix HuGeneFL GeneChip arrays. We first preprocessed the raw data using GCRMA with empirical Bayes estimate [44]. Then we filtered out probe sets which are either not annotated or have little variability across samples. Probes for 5669 genes were remained after preprocessing.

Our method can find differences between two tumour types at molecular level. We construct a classifier using the selected 20 samples with the accuracy of tumour type prediction approaching to 100% in the 10-fold cross validation procedure. The beauty of our method is that it does not only have strong predictive power, but also selects relevant features that could be candidates of disease biomarkers. The following parts of this section focus on investigating the performance of features selection. In the classification model of the whole dataset, 98 features have non-zero weights, which can be regarded as molecular features distinguishing tumours. By looking at Fig 3, we can see that many of these features are highly correlated, telling that our method does not discard features from correlated ones. We also apply SBL to

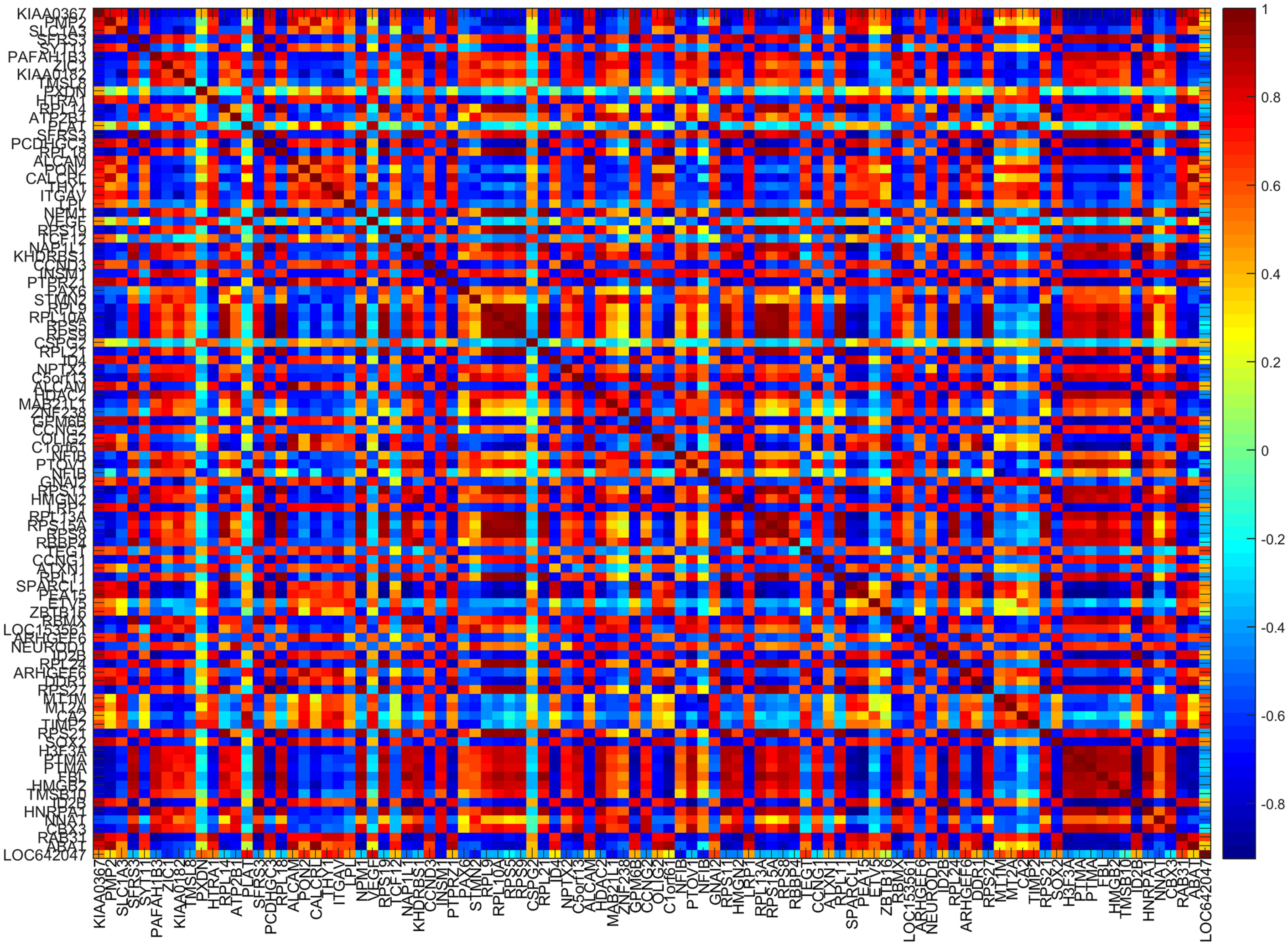

Fig 3. Heatmap of the correlation matrix for 98 genes selected by our classifier.

https://doi.org/10.1371/journal.pone.0189541.g003 construct a classification model. Although it can also return high predictive accuracy, it only selects two features with non-zero weights in the model. As RLR cannot work well for large datasets, we do not present its results for comparison.

The weights of features estimated from our method are compared with significant levels of features from the traditional statistics tests, which check features one by one to see whether the distributions of each feature in different groups are significantly different. Only the features whose p-values are smaller than the significant level (e.g., 0.05) are selected. Although there are many *p* value correction methods, a hard cut-off value is still needed. The number of selected features depends on the value of the significant level. Although it is conventional to set the significant level to be 0.05 or 0.01, we can hardly say any features whose p-value is slightly larger than this value do not have discriminant power. In this experiment, a t-test for each gene is conducted to find significant changes in expression levels between the MD and Mglio samples. We show the top 20 genes with the smallest *p* values in Fig 4.

Fig 4 compares the weights resulting from our method and SBL with fold changes and *p* values. Fig 4A) shows the weights of features in our classification model, where the circle points indicates the top 20 features from the *t* test. We can see that most features which are statistically significant have non-zero weights. In contrast to our method, Fig 4B) shows the results from SBL, telling that the top 20 features from the *t* test all have zero weights. From Fig 4A) and 4B) we can see our method outperforms SBL in the aspect of feature selection. Fig 4C) shows the fold change of the top 20 statistically significant features. Comparing Fig 4A) with 4C) we can see that the signs of weights agree with the signs of fold changes. It demonstrates that our method can reflect whether a gene is up-regulated or down-regulated for MD compared to Mglio. Fig 4D) shows the $\exp(-\log(\cdot))$ transformed *p* values for the top 20 statistically significant genes. It should be noted that although we compare our results with *p* value results, we cannot simply regard *p* value results as the ground truth. The disagreements between our method and *t* test can be resulted from the case that *t* test gives wrong results. This is because, in the statistic tests, features are investigated separately. It is often the case that some individual features are not discriminant but have strong predictive power when they join together. If we compare our results with the ground truth, we may find a better comparison results than those shown in Fig 4. As the ground truth is not at hand, we use *p* value results for comparison.

After comparing our results with other methods, we then investigate the biological functions of the 98 selected genes from our method. In this experiment, we determine a list of 53 up-regulated genes for MD compared to Mglio. Genes from this list are analysed for functional category enrichment using the Functional Annotation Clustering tool on the Database for Annotation, Visualisation and Integrated Discovery [45]. Metastasis-associated genes are classified according to their annotated role in molecular function, biological process, and cellular component from Gene Ontology (GO). Category enrichment is tested against all human genes, where *p* values are adjusted using the Benjamini-Hochberg multiple testing correction method [46]. In S2 Table, we show the discovered GO annotation clusters, where the detected GO terms are consistent with the findings in [8]. We can further select the GO terms of interest and build a sub-ontology that includes the ancestors of the terms. Fig 5 shows the ontology built using the top 10 significant GO terms. From this experiment, we can see that our method can work as a classifier and also a feature selection method whose output (selected features) can be fed into downstream analyses such as gene set enrichment and pathway analysis.

Discussion

With the development of high-throughput technology, biological process can be quantitatively measured. Differential feature detection and classification model construction are two main

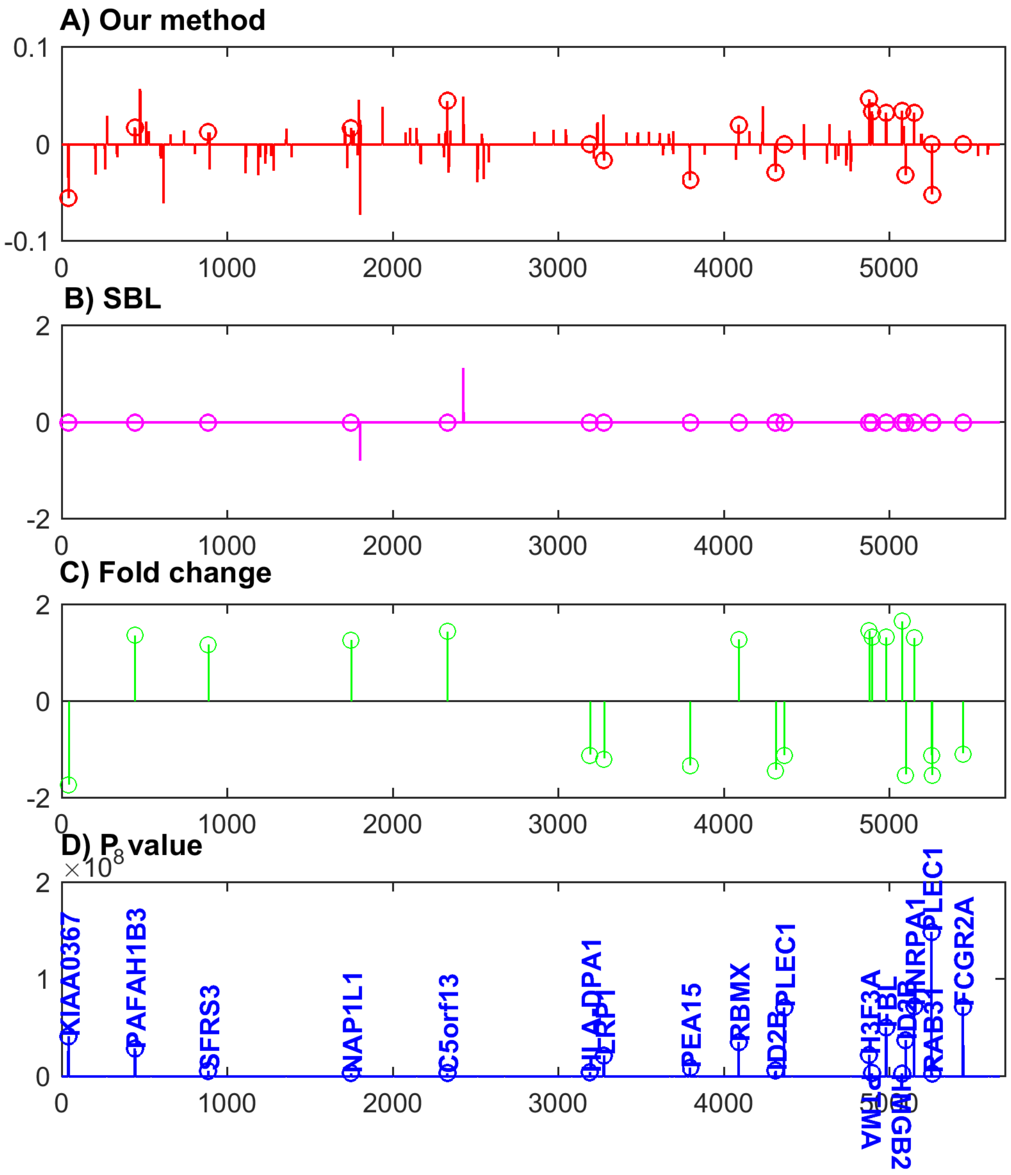

Fig 4. The comparisons for the estimated weights from our method and SBL with fold changes and *p* value. The top 20 features from *t* tests are shown as circle points. A) and B) are weights of features from the classification models using our method and SBL, respectively; C) shows fold changes of the top 20 statistically significant features; D) shows the exp(−log(·)) transformed *p* values for the top 20 statistically significant genes (labelled by official gene symbol).

https://doi.org/10.1371/journal.pone.0189541.g004

analyses in the biological expression study. In this paper, we propose a method to perform these two analyses at the same time: the model can predict sample groups; and the features used in the model with non-zero weights can be regarded as potential biomarkers. Currently, there are many machine learning methods for classification model construction. Most of them *cannot* directly return a list of predictive features with non-zero weights in the model. For example, the linear SVM may use all features for model construction, where they all have non-zero weights.

As SBL imposes sparsity to the model, a lot of features are forced to be pruned in the classification model. Thus, we use SBL as the basis of our method. Different from SBL which follows

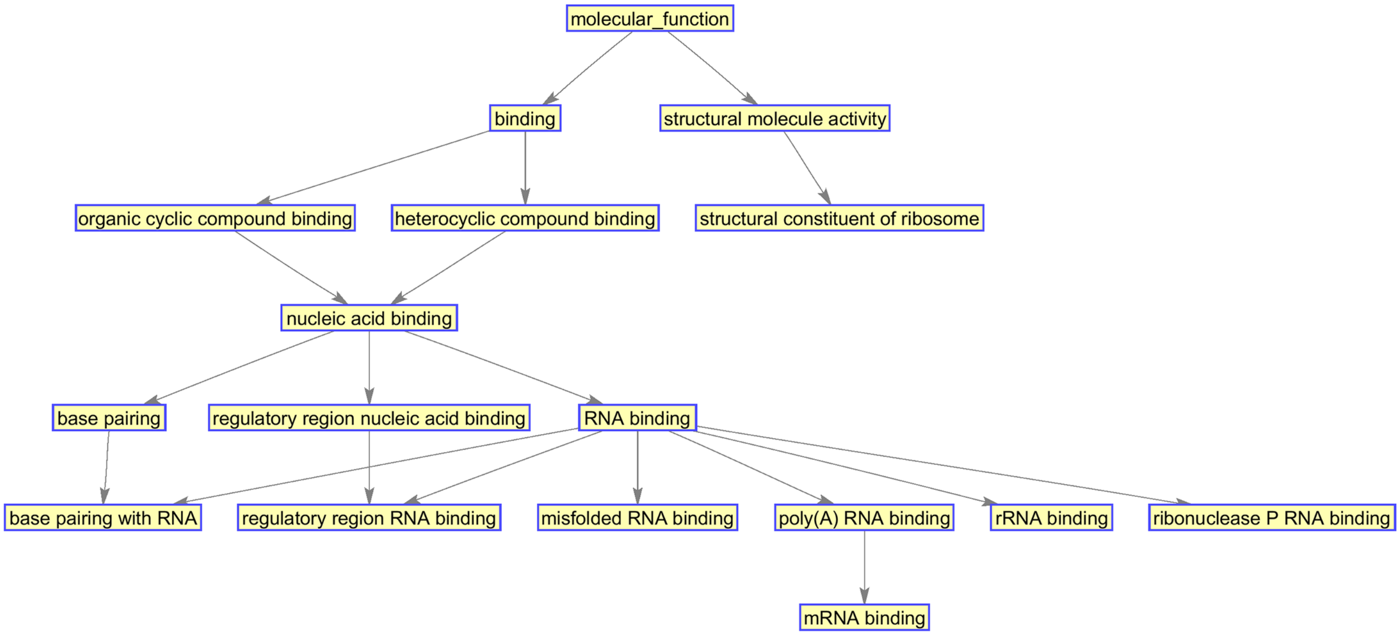
molecular_function
binding
structural molecule activity
organic cyclic compound binding
heterocyclic compound binding
structural constituent of ribosome
nucleic acid binding
base pairing
regulatory region nucleic acid binding
RNA binding
base pairing with RNA
regulatory region RNA binding
misfolded RNA binding
poly(A) RNA binding
rRNA binding
ribonuclease P RNA binding
mRNA binding

Fig 5. The ontology built using the top 10 significant GO terms discovered from up-regulated genes for MD compared to Mglio.

https://doi.org/10.1371/journal.pone.0189541.g005

an EM style to infer parameters and hyperparameters iteratively, we formulate the inference problem in the framework of optimisation: the target function in the optimisation process is originated from SBL; the iterative updating procedure follows the idea of convex-concave optimisation. Compared with SVM-based methods, our method has the following distinguishing features: 1) Our method is parameter free that hyperparameters are directly learned from datasets, while SVM-based methods need to set parameters through the cross validation process. 2) Our method imposes sparse constraints to the classification model. By choosing linear kernel, we can obtain a small set of features with non-zero weights used in model. The feature selection and classification steps have been integrated into one step. 3) Our method can detect correlated features which are important for downstream analysis, such as functional pathway analysis. Our method is also different from other optimisation based methods. Let us take the BPSO method as an example for discussion. The main differences between our method and BPSO are: 1) The BPSO method is inherently a global optimisation method. Our method although is an optimisation method, it constructs the model from the Bayesian point of view, where prior knowledge can be explicitly included in the model. 2) Our method is parameter free that hyperparameters are learned from the data. BPSO needs to set parameters in advance or obtained them via the cross validation process. 3) Looking at the results from BPSO, we can see that in each run of BPSO, only a small subset of genes is selected (e.g., 10 genes). 4) Our method carries out classification and feature selection in one step, where BPSO is mainly used for feature selection requiring other classification method such as ELM for classification model construction.

The simulation results show that our method can effectively select features with high classification accuracy. In contrast to other methods, correlated features can be successfully detected. A real gene expression data from the embryonal brain tumour study is then used to demonstrate the applicability of our method. In the results, we first show that the selected features are correlated by looking at the heatmap of correlation matrix. Then we compare the weights estimated from our method with p values from statistic test and fold changes. We find that our method can successfully identify up-regulated and down-regulated genes with positive and negative weights, respectively. Moreover, we find that most features which are statistically significant have non-zero weights in our model. The gene list generated by our method can be used to do functional analysis. We show the detected gene ontology terms, which are consistent with the findings in previous study. In conclusion, the classification and feature selection method proposed in this paper can effectively handle highly correlated biological expression dataset, in order to predict distinct disease subtypes and select candidates of biomarkers simultaneously.

Supporting information

S1 Table. The performance of our method, the regularized logistic regression method (RLR) and the SBL method with the sparsity of 0.2. The number of samples and features is chosen from 50, 100, 500. The percentage of correlated features (%cor) can be 0 or 50.
(XLS)

S2 Table. The discovered GO annotation clusters.
(XLS)

Author Contributions

Conceptualization: Xian Yang, Wei Pan, Yike Guo.

Formal analysis: Xian Yang.

Methodology: Xian Yang, Wei Pan, Yike Guo.

Project administration: Yike Guo.

Software: Xian Yang.

Supervision: Yike Guo.

Validation: Xian Yang.

Writing – original draft: Xian Yang.

Writing – review & editing: Xian Yang, Wei Pan, Yike Guo.

References

1. Schneider MV, Orchard S. Omics technologies, data and bioinformatics principles. Bioinformatics for Omics Data: Methods and Protocols. 2011; p. 3–30.
2. Alyass A, Turcotte M, Meyre D. From big data analysis to personalized medicine for all: challenges and opportunities. BMC medical genomics. 2015; 8(1):33. https://doi.org/10.1186/s12920-015-0108-y PMID: 26112054
3. Slonim DK, Yanai I. Getting started in gene expression microarray analysis. PLoS Comput Biol. 2009; 5 (10):e1000543. https://doi.org/10.1371/journal.pcbi.1000543 PMID: 19876380
4. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences. 2005; 102(43):15545–15550. https://doi.org/10.1073/pnas.0506580102
5. Slonim DK. From patterns to pathways: gene expression data analysis comes of age. Nature genetics. 2002; 32:502–508. https://doi.org/10.1038/ng1033 PMID: 12454645
6. Osareh A, Shadgar B. Classification and diagnostic prediction of cancers using gene microarray data analysis. Journal of Applied Sciences. 2009; 9(3):459–468. https://doi.org/10.3923/jas.2009.459.468

7. Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, et al. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature genetics. 2002; 30(1):41–47. https://doi.org/10.1038/ng765 PMID: 11731795

8. Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature. 2002; 415 (6870):436–442. https://doi.org/10.1038/415436a PMID: 11807556

9. Glinsky GV, Glinskii AB, Stephenson AJ, Hoffman RM, Gerald WL. Gene expression profiling predicts clinical outcome of prostate cancer. The Journal of clinical investigation. 2004; 113(6):913–923. https://doi.org/10.1172/JCI20032 PMID: 15067324

10. Hinton G, Deng L, Yu D, Dahl GE, Mohamed Ar, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine. 2012; 29(6):82–97. https://doi.org/10.1109/MSP.2012.2205597

11. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–1105.

12. Wainwright MJ, Jordan MI, et al. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning. 2008; 1(1–2):1–305.

13. Koller D, Friedman N. Probabilistic graphical models: principles and techniques. MIT press; 2009.

14. Zhu J, Chen N, Xing EP. Bayesian inference with posterior regularization and applications to infinite latent SVMs. Journal of Machine Learning Research. 2014; 15(1):1799–1847.

15. Ghahramani Z, Griffiths TL. Infinite latent feature models and the Indian buffet process. In: Advances in neural information processing systems; 2006. p. 475–482.

16. Teh YW, Jordan MI, Beal MJ, Blei DM. Sharing clusters among related groups: Hierarchical Dirichlet processes. In: Advances in neural information processing systems; 2005. p. 1385–1392.

17. Cogill S, Wang L. Support vector machine model of developmental brain gene expression data for prioritization of Autism risk gene candidates. Bioinformatics. 2016; 32(23):3611–3618. PMID: 27506227

18. Carlson TA, Schrater P, He S. Patterns of activity in the categorical representations of objects. Journal of cognitive neuroscience. 2003; 15(5):704–717. https://doi.org/10.1162/jocn.2003.15.5.704 PMID: 12965044

19. Mitchell TM, Hutchinson R, Niculescu RS, Pereira F, Wang X, Just M, et al. Learning to decode cognitive states from brain images. Machine learning. 2004; 57(1):145–175. https://doi.org/10.1023/B:MACH.0000035475.85309.1b

20. Liu B, Wu H, Zhang D, Wang X, Chou KC. Pse-Analysis: a python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods. Oncotarget. 2017; 8 (8):13338. https://doi.org/10.18632/oncotarget.14524 PMID: 28076851

21. Chang Cc, Lin H. A library for support vector machines. 2007;

22. Liu B, Long R, Chou KC. iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. Bioinformatics. 2016; 32(16):2411–2418. https://doi.org/10.1093/bioinformatics/btw186 PMID: 27153623

23. Tipping ME. Sparse Bayesian learning and the relevance vector machine. Journal of machine learning research. 2001; 1(Jun):211–244.

24. Pan W. Bayesian learning for nonlinear system identification. Imperial College London; 2015.

25. Lu C, Devos A, Suykens JA, Arús C, Van Huffel S. Bagging linear sparse Bayesian learning models for variable selection in cancer diagnosis. IEEE Transactions on Information Technology in Biomedicine. 2007; 11(3):338–347. https://doi.org/10.1109/TITB.2006.889702 PMID: 17521084

26. Eyheramendy S, Genkin A, Ju WH, Lewis DD, Madigan D. Sparse bayesian classifiers for text categorization. Journal of Intelligence Community Research and Development. 2003; 13.

27. Han F, Yang C, Wu YQ, Zhu JS, Ling QH, Song YQ, et al. A Gene Selection Method for Microarray Data Based on Binary PSO Encoding Gene-to-Class Sensitivity Information. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). 2017; 14(1):85–96. https://doi.org/10.1109/TCBB.2015.2465906

28. Wang L, Zhu J, Zou H. Hybrid huberized support vector machines for microarray classification and gene selection. Bioinformatics. 2008; 24(3):412–419. https://doi.org/10.1093/bioinformatics/btm579 PMID: 18175770

29. Zhang HH, Ahn J, Lin X, Park C. Gene selection using support vector machines with non-convex penalty. bioinformatics. 2006; 22(1):88–95. https://doi.org/10.1093/bioinformatics/bti736 PMID: 16249260

30. Shevade SK, Keerthi SS. A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics. 2003; 19(17):2246–2253. https://doi.org/10.1093/bioinformatics/btg308 PMID: 14630653

31. Huang HH, Liu XY, Liang Y. Feature Selection and Cancer Classification via Sparse Logistic Regression with the Hybrid L 1/2+ 2 Regularization. PloS one. 2016; 11(5):e0149675. https://doi.org/10.1371/journal.pone.0149675 PMID: 27136190

32. Ai-Jun Y, Xin-Yuan S. Bayesian variable selection for disease classification using gene expression data. Bioinformatics. 2010; 26(2):215–222. https://doi.org/10.1093/bioinformatics/btp638

33. Li Y, Campbell C, Tipping M. Bayesian automatic relevance determination algorithms for classifying gene expression data. Bioinformatics. 2002; 18(10):1332–1339. https://doi.org/10.1093/bioinformatics/18.10.1332 PMID: 12376377

34. Krishnapuram B, Carin L, Hartemink AJ. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. Journal of Computational Biology. 2004; 11(2-3):227–242. https://doi.org/10.1089/1066527041410463 PMID: 15285890

35. Cawley GC, Talbot NL. Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. Bioinformatics. 2006; 22(19):2348–2355. https://doi.org/10.1093/bioinformatics/btl386 PMID: 16844704

36. Xing EP, Ho Q, Xie P, Wei D. Strategies and principles of distributed machine learning on big data. Engineering. 2016; 2(2):179–195. https://doi.org/10.1016/J.ENG.2016.02.008

37. Yuille AL, Rangarajan A. The concave-convex procedure. Neural computation. 2003; 15(4):915–936. https://doi.org/10.1162/08997660360581958 PMID: 12689392

38. Pan W, Yuan Y, Gonçalves J, Stan Gb. A Sparse Bayesian Approach to the Identification of Nonlinear State-Space Systems. IEEE TRANSACTIONS ON AUTOMATIC CONTROL. 2016; 61(1):1–14. https://doi.org/10.1109/TAC.2015.2426291

39. Lipp T, Boyd S. Variations and extension of the convex–concave procedure. Optimization and Engineering. 2016; 17(2):263–287. https://doi.org/10.1007/s11081-015-9294-x

40. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Mathematical programming. 1989; 45(1):503–528. https://doi.org/10.1007/BF01589116

41. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning. 2011; 3 (1):1–122. https://doi.org/10.1561/2200000016

42. Yan S, Yang X, Wu C, Zheng Z, Guo Y. Balancing the stability and predictive performance for multivariate voxel selection in fMRI study. In: International Conference on Brain Informatics and Health. Springer; 2014. p. 90–99.

43. Kirk P, Witkover A, Bangham CR, Richardson S, Lewin AM, Stumpf MP. Balancing the robustness and predictive performance of biomarkers. Journal of Computational Biology. 2013; 20(12):979–989. https://doi.org/10.1089/cmb.2013.0018 PMID: 23909374

44. Wu Z, Irizarry RA, Gentleman R, Martinez-Murillo F, Spencer F. A model-based background adjustment for oligonucleotide expression arrays. Journal of the American statistical Association. 2004; 99 (468):909–917. https://doi.org/10.1198/016214504000000683

45. Huang DW, Sherman BT, Tan Q, Collins JR, Alvord WG, Roayaei J, et al. The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome biology. 2007; 8(9):R183. https://doi.org/10.1186/gb-2007-8-9-r183 PMID: 17784955

46. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the royal statistical society Series B (Methodological). 1995; p. 289–300.