SCIENTIFIC REPORTS
nature research

OPEN

Multi-view based integrative analysis of gene expression data for identifying biomarkers

Received: 8 April 2019
Accepted: 30 August 2019
Published online: 18 September 2019

Zi-Yi Yang[1], Xiao-Ying Liu[2], Jun Shu[3], Hui Zhang[1], Yan-Qiong Ren[1], Zong-Ben Xu[3] & Yong Liang[1]

The widespread applications in microarray technology have produced the vast quantity of publicly available gene expression datasets. However, analysis of gene expression data using biostatistics and machine learning approaches is a challenging task due to (1) high noise; (2) small sample size with high dimensionality; (3) batch effects and (4) low reproducibility of significant biomarkers. These issues reveal the complexity of gene expression data, thus significantly obstructing microarray technology in clinical applications. The integrative analysis offers an opportunity to address these issues and provides a more comprehensive understanding of the biological systems, but current methods have several limitations. This work leverages state of the art machine learning development for multiple gene expression datasets integration, classification and identification of significant biomarkers. We design a novel integrative framework, MVIAm - Multi-View based Integrative Analysis of microarray data for identifying biomarkers. It applies multiple cross-platform normalization methods to aggregate multiple datasets into a multi-view dataset and utilizes a robust learning mechanism Multi-View Self-Paced Learning (MVSPL) for gene selection in cancer classification problems. We demonstrate the capabilities of MVIAm using simulated data and studies of breast cancer and lung cancer, it can be applied flexibly and is an effective tool for facing the four challenges of gene expression data analysis. Our proposed model makes microarray integrative analysis more systematic and expands its range of applications.

Microarray technology is one of the most recent advances being used for cancer research, which can measure the expression levels of many thousands or tens of thousands of genes simultaneously. With the rapid development of microarray technology, many database repositories of high throughput gene expression data have been created and published for researchers to use, Gene Expression Omnibus (GEO), for example, currently have stored more than 2.76 million samples over 105,000 studies[1]. The use of gene expression datasets to discover highly reliable biomarkers is an important goal in clinical applications. The significant biomarkers can help researchers to detect the disease in individuals, classify the type of disease, predict the response of therapy and so on[2].

Analysis of gene expression data using biostatistics and machine learning approaches is facing four major challenges: (1) High noise: Random noise and systematic biases exist in gene expression data not only impact the scientific validity and costs of studies but also disrupts accurate prediction of phenotype that may ultimately impact patients[3,4]. (2) Small sample size with high dimensionality: The gene expression dataset generally contains a large number of genes and small size of samples, which called large p & small n problem[5]. Only a small fraction of genes are closely relevant to the target disease, and most genes are irrelevant[6]. From a machine learning perspective, numerous irrelevant genes may introduce noise and reduce the performance of the classifier[7,8]. (3) Batch effects: It occurs because measurements are affected by many factors including experiments principle, data collection standards, and personnel differences. The systematic noise introduced when samples are processed in multiple batches have a detrimental effect on data derived from microarrays[9,10]. (4) Low reproducibility of significant biomarkers: The published significant biomarkers from internal validation rarely overlap with other research groups[11]. These four issues reveal the complexity of gene expression data, which constrains the development of microarray technology in clinical applications.

[1]Faculty of Information Technology & State Key Laboratory of Quality Research in Chinese Medicines, Macau University of Science and Technology, Taipa, 999078, Macau, China. [2]Computer Engineering Technical College, Guangdong Polytechnic of Science and Technology, Zhuhai, 519090, China. [3]School of Mathematics and Statistics & Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, 710049, China. Correspondence and requests for materials should be addressed to Y.L. (email: yliang@must.edu.mo)

To face these challenges and take advantage of multiple published gene expression datasets, the integrative analysis of gene expression data has become an effective tool by aggregating multiple datasets and increasing the statistical power in identifying a small subset of genes to effectively predict the type of the disease[12,13]. Current microarray integrative analysis was first proposed by Hamid *et al.*[14], basically classified into "late stage" data integration and "early stage" data integration. However, current methods for microarray integrative analysis have several limitations. Most "late stage" data integration methods identify genes based on combining univariate summary statistics, such as *p*-value[15], effect size[16] and rank aggregation[12,17]. As a result, it is difficult to identify non-redundant significant genes and systematically determine (e.g. cross-validation) how many genes to include in the subset, such as GeneMeta[18] and metaArray[19]. Moreover, such methods neglect correlations among genes and do not eliminate the batch effects between different datasets. Current "early stage" data integration methods usually apply one cross-platform normalization method to aggregate multiple datasets into a single unified large dataset. After that, classification and variable selection for the merged dataset can be achieved by the machine learning methods. For example, Ma *et al.*[20] proposed the meta threshold gradient descent regularization (MTGDR) for gene selection in the integrative analysis of gene expression data. Meta-lasso method was published by Li *et al.*[21], which not only boosts the statistic power to identify significant genes but also keeps the flexibility of gene selection. Recently, Hughey *et al.*[22] developed integrative analysis using elastic net penalized with logistic regression model (L_{EN}), a powerful and versatile method for variable selection in classification. Special emphasis, cross-platform normalization is an essential part of the "early stage" data integration, because it can eliminate the differences between datasets from different microarray platforms while preserving underlying the differences in biology[23]. A number of cross-platform normalization methods have been developed and provide effective batch adjustment for microarray data, such as ComBat[24], cross-platform normalization (XPN) method[25], and batch effects removal (ber)[26]. However, different cross-platform normalization methods are based on different statistical models with different accuracy, precision and overall effectiveness[27]. Current "early stage" data integration methods usually apply one cross-platform normalization method, which cannot ensure maximum elimination of the batch effects. Beyond that, none of these integrative analysis methods have a robust learning mechanism to minimize the influence of the noise. Therefore, there is a crucial need for a novel integrative analysis method for robust analysis of the microarray data, prediction of cancer types and identification of significant biomarkers.

We design a novel integrative framework called MVIAm (Multi-View based Integrative Analysis of microarray data for identifying biomarkers). MVIAm can be divided into three phases: pre-processing each dataset, aggregation and generate multi-view data, and analysis of multi-view data. MVIAm aggregates multiple microarray gene expression datasets through different cross-platform normalization methods and generates multiple aggregated gene expression datasets. Each aggregated dataset has the same set of samples and features but is generated by the different statistical models, which belongs to one type of multi-view data[28]. The novel integrative framework MVIAm extends the traditional "early" stage data integration to multi-view data integration. Generally, multi-view data contains complementary information and has more comprehensive information than those of single-view data[29]. In recent years, several multi-view machine learning methods for integrating multi-view data have been developed[28,30]. The supervised multi-view data integration methods generally include concatenation-based and ensemble-based integration[31]. MVIAm enables more multi-view machine learning methods for supervised homogeneous data integration. The multi-view gene expression data generated by MVIAm has the following characteristics:

Multi-view data generated by MVIAm can significantly increase the sample size, which greatly alleviates large *p* & *n* problem and increase the statistical power in identifying biomarkers.

- Multi-view data typically contains complementary information and has more comprehensive understanding of the biological systems.
- The batch effects cannot be completely eliminated, meaning that each view of the data still has different types of bias.

Although quality control and different cross-platform normalization methods are used to process gene expression data, it is inevitable that the data has noises and biases. In the phase of analyzing gene expression data, in order to alleviate the impact of the noise on the learning process and take advantage of significantly increased data, we introduce a robust learning mechanism called self-paced learning[32]. Self-paced learning (SPL) is a typical sample reweighting method, especially used in high noise situations[33]. It was proposed based on the core idea of curriculum learning[34]. Curriculum learning (CL) is inspired by human learning and is learned by gradually including samples from easy to complex into the training process. SPL embeds curriculum design as a regularization term into the learning objective, automatically select samples into training from easy to complex in a purely self-paced way. Due to its generality and generalization, SPL has been widely used in various tasks[35–38]. Moreover, Meng *et al.*[39] have provided some new theoretical understanding of the SPL scheme, which helps us have a deep insight into it. To analysis multi-view gene expression data, we propose Multi-View Self-Paced Learning (MVSPL), a robust supervised multi-view data integration method. The main idea of MVSPL is to interactively recommend high-confidence samples with smaller loss values and automatically select samples from easy to complex to train the model for each view.

In summary, the main contributions of this work can be summarized as follows:

- We design a novel framework of gene expression data integration called MVIAm, which can generate multi-view gene expression data based on different cross-platform normalization methods. Moreover, we propose a robust learning method MVSPL to analyze multi-view gene expression data for gene selection and cancer classification problem. It is an effective tool to address the challenges of microarray data analysis.

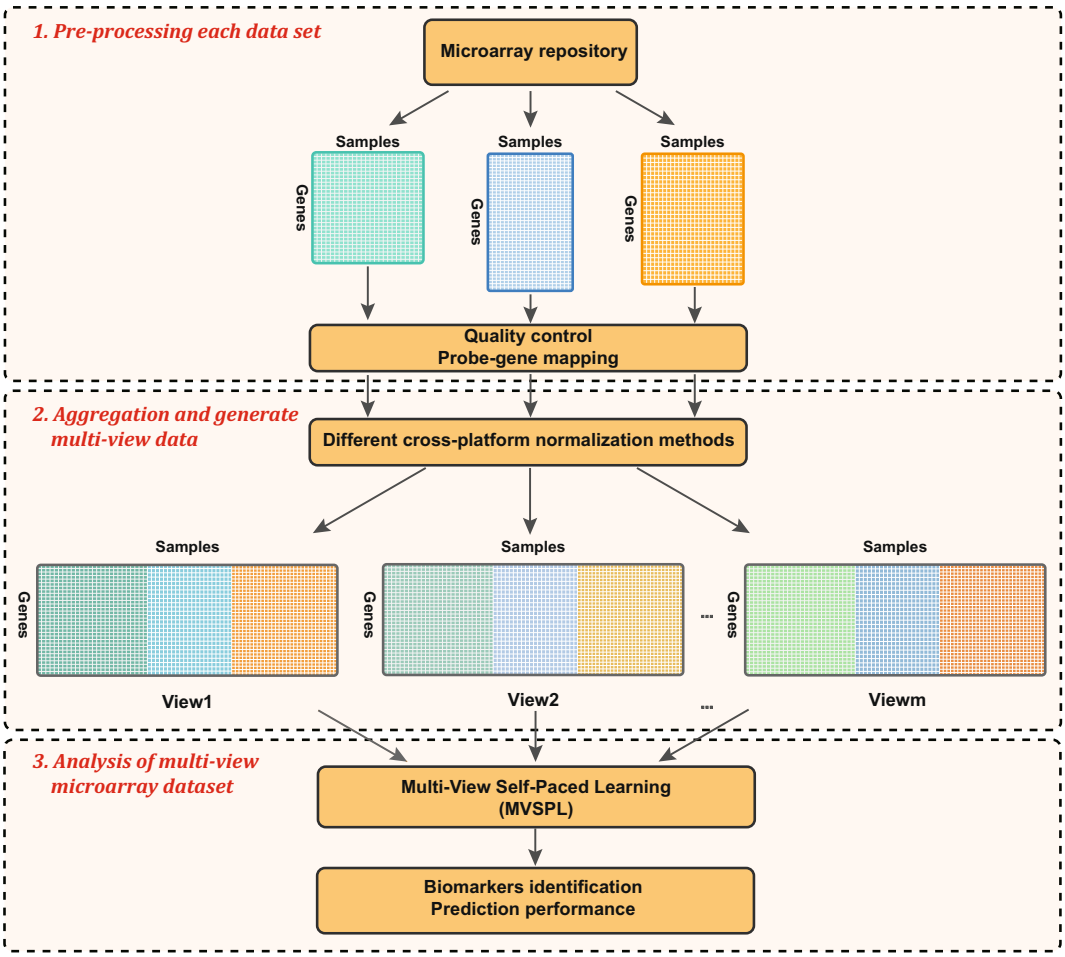

Figure 1. MVIAm, a novel framework for data integrative analysis. The first phase inputs multiple microarray datasets and processes the data according to the pre-processing steps. For the second phase of MVIAm, it applies multiple cross-platform normalization methods to aggregate multiple datasets. Each aggregated dataset possesses the same set of samples and genes, but it is generated by the different statistical normalization models, which belongs to one type of multi-view data. The third phase is the analysis of multi-view microarray data, we propose the MVSPL approach to identify significant biomarkers and predict the type of cancer.

- Experimental results on both simulation and real experiments substantiate the superiority of MVSPL as compared to a sparse logistic regression model with Lasso (L_1), a sparse logistic regression model with elastic net (L_{EN}), ensemble-based elastic net (Ensemble_EN) and SPL.
- Our proposed model makes gene expression integrative analysis more systematic and expands the range of applications that an integrative analysis can be used to address.

Methods

The MVIAm integrative framework. Figure 1 shows the pipeline of the MVIAm, which aggregates multiple microarray datasets and identifies the significant biomarkers, assesses the prediction performance of the model. MVIAm can be divided into three phases: pre-processing each dataset, aggregation and generate multi-view data, and analysis of multi-view data.

Pre-processing each data set. The original Affymetrix data was first normalized and log-transformed by a robust multi-array average (RMA)[40] method. After that, downloading and installing the appropriate custom chip definition files (CDFs) packages according to the type of microarray platform. The CDF package is necessary for probe annotation for Affymetrix data. The probes of the normalized data can be successfully mapped to Entrez Gene IDs by annotation packages in Bioconductor[41]. If multiple probes match a single Entrez ID, we calculated the median of values of those probes as the expression value for this gene.

Aggregation and generate multi-view data. One challenge of microarray integrative analysis is that each gene expression dataset may have gene expression values for slightly different sets of genes. Commonly method, the common genes from all gene expression datasets are extracted as the merged set of genes. After that, MVIAm utilizes different cross-platform normalization methods to process the gene expression dataset to eliminate the batch

effects. In this work, we use two cross-platform normalization methods to eliminate the batch effects, ComBat[24] and ber[26]. ComBat is an Empirical Bayes method, includes two methods, a parametric prior method (ComBat_p) and a non-parametric method (ComBat_n), based on the prior distributions of the estimated parameters. Ber, removes batch effects by using a two-stage regression approach, includes two methods, with bagging method (ber_bg) and without bagging method (ber).

Multi-view self-paced learning (MVSPL). Here, we detailed introduce the proposed multi-view self-paced learning (MVSPL) model, which extends the self-paced learning[35] model to multi-view scenarios. The fundamental concept of SPL please see the part of related work. Suppose given a dataset with multiple views $D = \{(X_1^{(j)}, y_1), (X_2^{(j)}, y_2), \ldots, (X_n^{(j)}, y_n)\}$, where $X_i^{(j)} = \left(x_{i1}^{(j)}, x_{i2}^{(j)}, \ldots, x_{ip}^{(j)}\right)$ is the i-th input sample with p features under the j-th view and y_i is the i-th sample with the value 0 or 1 for every view in the classification model. Let $L(y_i, f(x_i^{(j)}, \beta^{(j)}))$ denotes the loss function, which calculates the loss between the real label y_i and the estimated value $f(x_i^{(j)}, \beta^{(j)})$ in the j-th view. The $\beta^{(j)}$ represents the model parameter inside the decision function $f(x_i^{(j)}, \beta^{(j)})$. The objective function of MVSPL can be expressed as:

$$\begin{aligned}\min_{\beta^{(j)}, v^{(j)} \in [0,1]^n, j=1,2,\ldots,m} E(\beta^{(j)}, v^{(j)}; \lambda^{(j)}, \gamma^{(j)}, \delta) \;=\; & \sum_{j=1}^{m}\sum_{i=1}^{n} v_i^{(j)} L(y_i, f^{(j)}(x_i^{(j)}, \beta^{(j)})) \\ & + \sum_{j=1}^{m} \lambda^{(j)} \|\beta^{(j)}\|_1 - \sum_{j=1}^{m}\sum_{i=1}^{n} \gamma^{(j)} v_i^{(j)} \\ & - \delta \sum_{\substack{1 \le k,j \le m, \\ k \ne j}} (v^{(k)})^T v^{(j)},\end{aligned} \tag{1}$$

where m denotes the total number of views. $x_i^{(j)}$ is the i-th input sample ($i = 1, 2, \ldots, n$) under the j-th view, and y_i is the corresponding label of $x_i^{(j)}$ for every j. $v_i^{(j)}$ denotes the weight of $x_i^{(j)}$. $\lambda^{(j)}$ is a tuning parameter in the j-th view, it controls the complexity of the model. $\gamma^{(j)}$ denotes the age parameter, which controls the learning pace in each iteration in the j-th view. δ is the parameter controls influence from other views when one view is going to select more training samples.

MVSPL actually corresponds to the sum of SPL model under multiple views plus a regularization term $\sum_{\substack{1 \le k,j \le m \\ k \ne j}} (v^{(k)})^T v^{(j)}$. This inner product encodes the relationship between multiple views. This new regularizer demonstrates the basic assumption that multi-view data usually contains complementary information and have more comprehensive information than those of single-view data. Therefore, this new regularizer enforces the weight penalizing the loss of one view similar to that of other views.

The alternative optimization strategy. The alternative optimization strategy (AOS) can be used to solve the MVSPL model. The optimization process is as follows:

Initialization. $v^{(1)}, v^{(2)}, \ldots, v^{(m)}$ are zero vectors in R^m. $\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(m)}$ are initialized with small values to allow a few samples into training for the first iteration. δ is set as a specific value in the whole learning process. Multiple classifiers are simultaneously trained on all samples in different views to obtain an initial loss of all samples in each view.

Update $v_i^{(k)}$ ($k = 1, 2, \ldots, m; k \ne j$). The purpose of this step is to prepare confident samples with non-zeros $v_i^{(k)}$ values for training on the j-th view. By calculating the derivative of Eq. (1) with respect to $v_i^{(k)}$, then we can obtain:

$$\frac{\partial E}{\partial v_i^{(k)}} = L_i(y_i, f^{(k)}(x_i^{(k)}, \beta^{(k)})) - \gamma^{(k)} - \delta \sum_{1 \le j \le m, j \ne k} v_i^{(j)}. \tag{2}$$

According to Eq. (2), we can obtain the optimal weight for the i-th sample in the k-th view:

$$v_i^{(k)} = \begin{cases} 1, & L_i(y_i, f^{(k)}(x_i^{(k)}, \beta^{(k)})) < \gamma^{(k)} + \delta \sum_{1 \le j \le m, j \ne k} v_i^{(j)}, \\ 0, & otherwise. \end{cases} \tag{3}$$

Update $v_i^{(j)}$. This step aims to define which samples will be selected into the training of the j-th view. The optimization process for the $v_i^{(j)}$ is the same as the previous step, expressed as:

$$v_i^{(j)} = \begin{cases} 1, & L_i(y_i, f^{(j)}(x_i^{(j)}, \beta^{(j)})) < \gamma^{(j)} + \delta \sum_{1 \le k \le m, k \ne j} v_i^{(k)}, \\ 0, & otherwise. \end{cases} \tag{4}$$

The difference is that the samples selected in this step will be directly used for training in the j-th view. Furthermore, we can easily observe that samples selected by other views possess higher probabilities than others to be selected into training.

Update $\beta^{(j)}$. The purpose of this step is to obtain the optimal solution for the j-th view. Here, we choose the logistic regression classifier to train the model. Equation (1) degenerates into penalized logistic regression optimization problem:

$$\min_{\beta^{(j)}} \sum_{i=1}^{n} v_i^{(j)} L_i(y_i, f^{(j)}(x_i^{(j)}, \beta^{(j)})) + \lambda^{(j)} \|\beta^{(j)}\|_1. \tag{5}$$

This problem can be readily solved by R package glmnet[42].

Age parameter $\gamma^{(j)}(j = 1, 2, \ldots, m)$ is increased to allow more samples with larger loss values into training in the next iteration. When $\gamma^{(j)}$ is small, only select easy samples under j-th view with small losses. With the growth of the $\gamma^{(j)}$, more samples under j-th view with larger losses will be gradually selected to train a more "mature" model. Then we repeat the above optimization process with respect to each variable under the different views until the maximum iteration times is reached.

The pipeline of the proposed MVSPL is shown in Supplementary Fig. S1. And the whole process of this alternative optimization strategy for solving MVSPL is summarized in Algorithm 1.

Algorithm 1. The alternative optimization strategy for solving MVSPL model.

1: **Input**: samples $x_1^{(1)}, \ldots, x_n^{(1)}, \ldots, x_1^{(m)}, \ldots, x_n^{(m)}$, labels $y_1, \ldots, y_n$, parameters $\gamma^{(1)}, \ldots, \gamma^{(m)}$, δ and max_iter.
2: **Output**: $\beta^{(1)}, \ldots, \beta^{(m)}$.
3: Initialize $v^{(1)}, \ldots, v^{(m)}$, and $\gamma^{(1)}, \ldots, \gamma^{(m)}$
4: Update $\beta^{(1)}, \ldots, \beta^{(m)}$
5: iter = 1
6: **while** iter < max_iter **do**
7: **for** $j \leftarrow 1$ to m **do**
8: **for** $k \leftarrow 1$ to m **and** $k \neq j$ **do**
9: Update $v^{(k)}$ based on Equation (3) : Prepare confident samples with non-zeros $v^{(k)}$ values for training on the j-th view
10: **end for**
11: Update $v^{(j)}$ based on Equation (4): Add samples into training of the j-th view
12: Update $\beta^{(j)}$ based on Equation (5): Train a classifier (logistic regression model for instance) of the j-th view
13: **end for**
14: Augment $\gamma^{(1)}, ..., \gamma^{(m)}$
15: iter $\leftarrow$ iter + 1
16: **end while**
17: Return $\beta^{(1)}, \ldots, \beta^{(m)}$

According to Algorithm 1, the MVSPL model can obtain the optimal solution for each view. Algorithm 1 jointly learns the modal parameter $\beta^{(j)}$ and the latent weight variables $v^{(j)}$, where $j = 1, \ldots, m$. Steps 7–11 compute the latent weight variables of all samples n in multiple views m with the time complexity of $O(n \times m^2)$. With the latent weight variables fixed, Step 12 computes the optimal solution based on the generalized linear model with lasso penalty by using Coordinate Descent algorithm[42] with the time complexity of $O(n^2 \times p)$, where p represents the number of features and $n \ll p$. This step computes the optimal solution in multiple views, so the time complexity is $O(n^2 \times p \times m)$. Due to $m \ll n$, therefore, the time complexity of Algorithm 1 is $O(n^2 \times p \times m)$.

In the test phase, when the test dataset $D' = \{X_1, X_2, \ldots, X_u\}$ with multiple views $(1, 2, \ldots, m)$ are coming, where u is the number of test samples. We first fix $\beta^{(1)}, \beta^{(2)}, \ldots, \beta^{(m)}$, and then predict the optimal y_k by solving the following minimization problem:

$$y_k = \underset{y_k}{argmin} \sum_{j=1}^{m} L_k(y_k, f^{(j)}(x_k^{(j)}, \beta^{(j)})) \tag{6}$$

Related work. *Self-paced learning (SPL).* The self-paced learning model combines a weighted loss term for all samples and a general self-paced regularizer imposed on the samples weight. Suppose given a dataset $D = \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is the i-th input sample with p features and y_i is class of the i-th sample (e.g. $y_i \in \{0, 1\}$). Let $L(y_i, f(x_i, \beta))$ denotes the loss function, which calculates the loss between the real label y_i and the estimated value $f(x_i, \beta)$. The β represents the model parameter inside the decision function $f(x_i, \beta)$. The goal of the SPL is to jointly learn the model parameter β and the latent weight variable $v = [v_1, v_2, \ldots, v_n]$ by minimizing:

$$\min_{\beta, v \in [0,1]^n} E(\beta, v; \lambda, \gamma) = \sum_{i=1}^{n} v_i L(y_i, f(x_i, \beta)) - \gamma \sum_{i=1}^{n} v_i + \lambda \|\beta\|_1 \tag{7}$$

where γ is the age parameter for controlling the learning pace and λ is a tuning parameter. The alternative optimization strategy algorithm can effectively solve the SPL problem. When β is fixed, the optimum weight variable $v^* = [v_1^*, v_2^{**}, \ldots, v_n^*]$ can be calculated by:

$$v_i^* = \begin{cases} 1, & L(y_i, f(x_i, \beta)) < \gamma \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

By jointly updating model parameter β and the latent weight variable v, we can conclude that: (1) When updating v with a fixed β, if the loss value of a sample is smaller than the age parameter γ, then the sample is treated as an easy sample with $v_i^* = 1$, otherwise, $v_i^* = 0$. (2) When updating β with a fixed v, using the selected samples ($v_i^* = 1$) to train the classifier. (3) Before running the next iteration, increase the age parameter γ to adjust the learning pace. When γ is small, only select easy samples with small loss values. With γ increases, more samples with larger losses will be gradually selected to train a more "mature" model.

By jointly learning the model parameter β and the latent weight variable v based on the iterative algorithm with gradually increasing the age parameter, more samples can be automatically selected into training from easy to complex in a self-paced way.

Results

We demonstrate the performance of the proposed MVSPL in simulation and real microarray experiments. Four methods are compared with the MVSPL method: Sparse logistic regression with the Lasso penalty (L_1)[43], Sparse logistic regression with the elastic net penalty (L_{EN})[44], Ensemble-based elastic net (Ensemble_EN)[45] and SPL[32]. When MVIAm generates single-view data, it degenerates into traditional "early stage" data integration, and data analysis can be performed by L_1, L_{EN} and SPL. Ensemble_EN constructs a prediction model on each view of data before combing the model predictions and obtains the final prediction result based on Eq. (6).

Analysis of simulated data. We generate three independent simulated datasets for integration and each dataset with the character of small sample size and high dimensionality. Using the normal distribution to generate $X = (X_1, X_2, \ldots, X_n)$ with n samples and each samples with p features, for the i-th sample, $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$. After that, the correlation parameter ρ can be added to the simulated data[46].

$$x_{ij} = z_{ij}\sqrt{1-\rho} + z_{i1}\sqrt{\rho},\ i \sim (1, \ldots, n),\ j \sim (2, \ldots, p). \tag{9}$$

where $z_{ij} \sim_{i.i.d.} N(0, 1)$. The simulated dataset is generated from the logistic regression model, which can be given as:

$$log\left(\frac{y_i}{1-y_i}\right) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \sigma \cdot \varepsilon, \tag{10}$$

where $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^T$ is the independent random errors from $N(0, 1)$, σ is the noise control parameter.

We generated simulated data by the above procedure. Three independent simulated datasets were generated with the same number of variables ($p = 2000$). The coefficient β is set as follows:

$$\beta = (\underbrace{1.5, -1.2, 1.8, -2, 2.5, -1.2, 1, -1.5, 2, -1.6}_{10}, \underbrace{0, \cdots, 0}_{1990}). \tag{11}$$

Four scenarios were designed for the simulated experiment:

Scenario 1: The sample size $n_{dataset1} = 100$, $n_{dataset2} = 100$ and $n_{dataset3} = 100$, the correlation coefficient $\rho = 0, 0.2, 0.4, 0.6$ and 0.8, the noise control parameter $\sigma = 0$.

Scenario 2: The sample size $n_{dataset1} = 100$, $n_{dataset2} = 100$ and $n_{dataset3} = 100$, the noise control parameter $\sigma = 0, 0.2, 0.4, 0.6$ and 0.8, the correlation coefficient $\rho = 0$.

Scenario 3: The sample size $n_{dataset1} = 50$, $n_{dataset2} = 100$ and $n_{dataset3} = 150$, the noise control parameter $\sigma = 0, 0.4$ and 0.8, the correlation coefficient $\rho = 0$.

Scenario 4: The sample size $n_{dataset1} = 100$, $n_{dataset2} = 100$ and $n_{dataset3} = 100$, the noise control parameter $\sigma_{dataset1} = 0.1$, $\sigma_{dataset2} = 0.2$ and $\sigma_{dataset3} = 0.3$, the correlation coefficient $\rho = 0.2$.

Three independent simulated datasets are processed based on MVIAm and aggregated into a large multi-view dataset. We use four functions ComBat_p, ComBat_n, ber and ber_bg to eliminate batch effects and generate view1, view2, view3 and view4 of the aggregated multi-view data, respectively. L_1, L_{EN} and SPL achieve the best performance in the view of data by using ComBat_p to eliminate the batch effects. Therefore, these three competing methods use the view1 of the aggregated dataset for data analysis in four scenarios. The proposed MVSPL and Ensemble_EN have the flexibility to analyze data in multiple views. In Scenarios 1, 2 and 3, MVSPL and Ensemble_EN perform data analysis through two views of data: view1 and view2. In Scenario 4, we further explore our proposed method and its flexible scalability. Perform MVSPL through the interaction of two views, three views and four views of data, respectively. In the simulated experiment, we first combine independent simulated datasets into a large aggregated dataset. Then, the aggregated dataset is divided into two groups with random sampling, 70% samples for training and remaining samples for testing. The estimation of the optimal regularization parameter λ of the training dataset is obtained by 10-fold cross-validation. We repeat this procedure 30 times and report the average measurement.

To evaluate the prediction performance of classifiers, the accuracy, sensitivity, specificity and AUC are used in the simulation and real experiments. The definitions of these evaluation indicators can refer to[47,48]. In addition, the evaluation indicators for variable selection are defined as follows[49]:

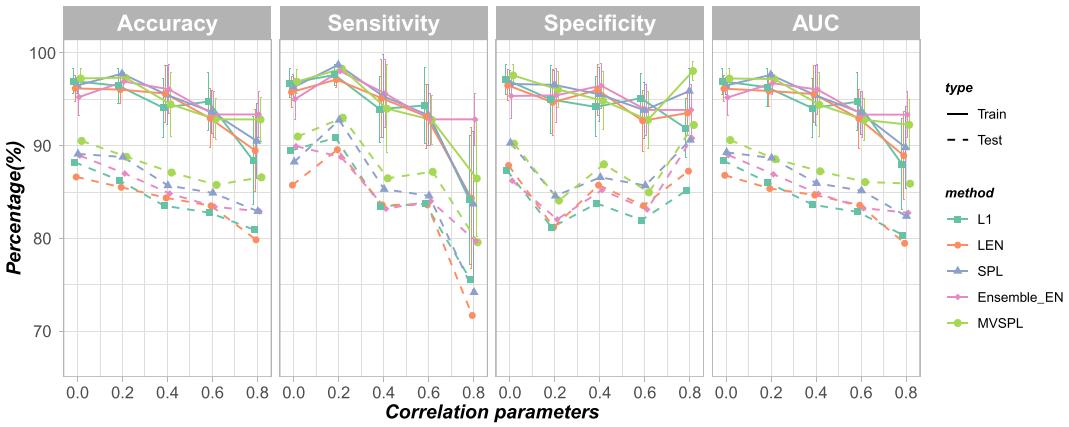

Figure 2. Prediction performance of the different methods with different correlation coefficient parameters. The error bars represent the standard deviation (SD).

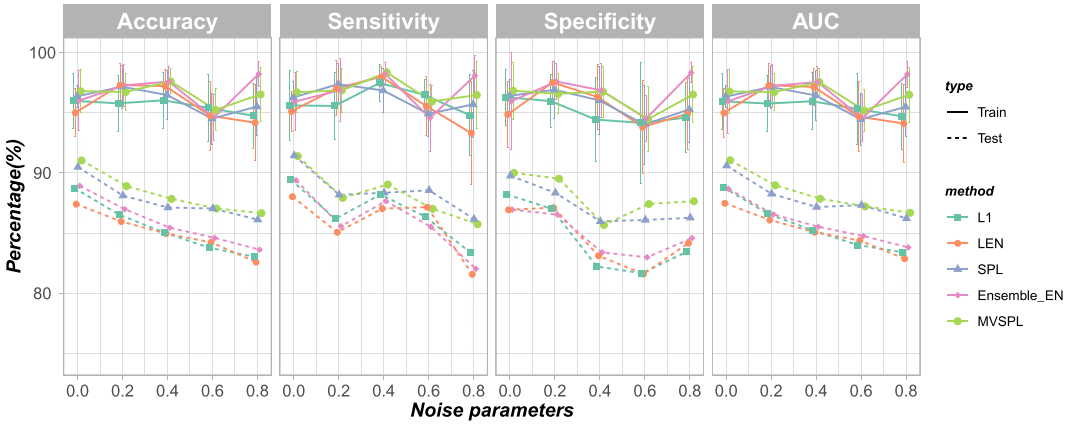

Figure 3. Prediction performance of the different integrative analysis methods with different noise control parameters. The error bars represent the standard deviation (SD).

Method	Correlation coefficient parameters					Noise control parameters				
	$\rho=0$	$\rho=0.2$	$\rho=0.4$	$\rho=0.6$	$\rho=0.8$	$\sigma=0$	$\sigma=0.2$	$\sigma=0.4$	$\sigma=0.6$	$\sigma=0.8$
	β-sensitivity									
L_1	91.21	94.85	88.79	82.24	66.67	90.48	90.82	91.12	90.52	88.18
L_{EN}	90.91	93.84	88.42	82.33	67.58	90.27	91.24	91.94	89.70	88.91
SPL	90.91	94.67	88.48	83.19	67.64	91.52	92.94	90.23	90.61	88.48
Ensemble_EN	89.67	93.67	92.33	87.67	68.33	89.67	92.33	91.67	90.67	90.34
MVSPL	**92.73**	**95.45**	**92.73**	**88.18**	**69.54**	**92.73**	**93.73**	**92.18**	**91.73**	**91.09**
	β-specificity									
L_1	98.71	98.71	98.79	98.46	98.87	98.81	98.79	98.68	98.56	98.63
L_{EN}	98.82	98.79	98.51	98.92	98.24	98.98	98.11	98.35	98.75	98.98
SPL	98.71	98.46	98.32	98.72	98.00	98.66	97.96	98.42	98.55	98.62
Ensemble_EN	98.77	98.20	98.01	98.49	97.90	98.54	97.86	97.44	98.55	96.94
MVSPL	98.42	98.44	97.86	98.16	98.37	98.50	97.49	98.02	97.75	97.01

Table 1. Variable selection performance (%) of the different integrative analysis methods with different parameters. The mean variable selection performance over 30 repetitions of the simulated experiments in Scenarios 1 and 2 are reported, and the best β-sensitivity are highlighted in bold.

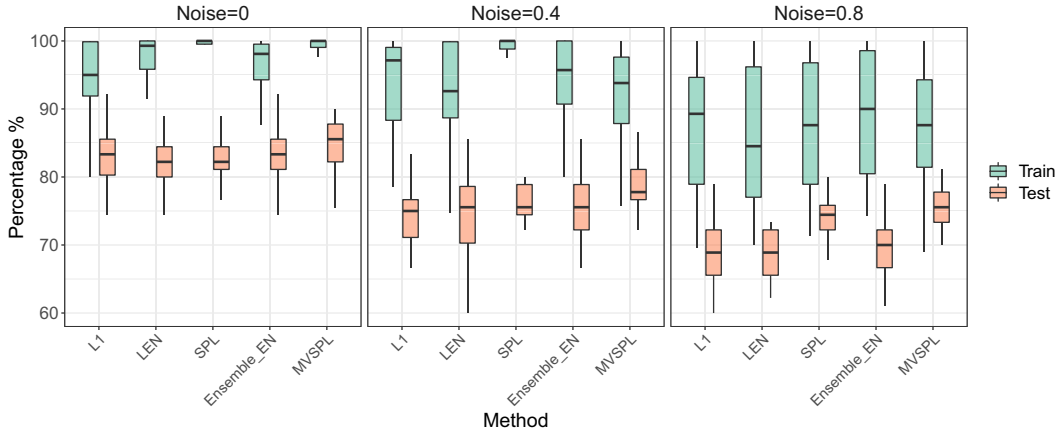

Figure 4. Boxplot diagram of training and test accuracy for the different methods with 30 repetitions in Scenario 3.

Dataset	No. of Probes	Classes (Class1/ Class2)	No. of Classes (Class1/ Class2)	Affymetrix Platform
GSE1561	22215	−ve/+ve	49 (22/27)	HG-U133A
GSE6532	22283	−ve/+ve	125 (40/85)	HG-U133A
GSE20437	22283	−ve/+ve	18 (9/9)	HG-U133A
GSE22093	22283	−ve/+ve	82 (41/41)	HG-U133A

Table 2. Four publicly available breast cancer gene expression datasets used in the real data experiments.

$$\begin{aligned} TruePositive(TP) &= |\beta. * \hat{\beta}|_0, TrueNegative(TN) = |\overline{\beta}. * \overline{\hat{\beta}}|_0 \\ FalsePositive(FP) &= |\overline{\beta}. * \hat{\beta}|_0, FalseNegative(FN) = |\beta. * \overline{\hat{\beta}}|_0 \\ \beta - sensitivity &= \frac{TP}{TP + FN}, \beta - specificity = \frac{TN}{TN + FP} \end{aligned} \tag{12}$$

where the $|\cdot|_0$ represents the number of non-zero elements in a vector. The logical not operators of β and $\hat{\beta}$ are $\overline{\beta}$ and $\overline{\hat{\beta}}$, respectively. And.* is the element-wise product.

In Scenario 1, we explored the effect of different correlation coefficient parameters on the performance of the five methods. As shown in Fig. 2, for the training dataset, the difference in prediction performance of all the methods is quite small. For the test dataset, it can be clearly seen that as the correlation parameter ρ increases, the prediction accuracy of all the five methods are decreased, expect for MVSPL in $\rho = 0.8$. The generalization ability of MVSPL and SPL are obviously superior to L_1, L_{EN} and Ensemble_EN. The average test accuracy, sensitivity, and AUC obtained by MVSPL are higher than the other competing methods with varying correlation coefficient parameters ρ. The results obtained by SPL are slightly inferior to MVSPL but better than the other three methods in most situations. Moreover, Ensemble_EN outperforms L_1 and L_{EN} with varying correlation parameters.

In Scenario 2, we explored the effect of different noise control parameters on the performance of the five methods. As shown in Fig. 3, consistent with the results of Scenario 1, all methods with the similar prediction performance in the training dataset. For the test dataset, when the noise control parameter increases, the prediction accuracy of all the competing methods are decreased. MVSPL and SPL demonstrate the excellent generalization performance. The average test accuracy and AUC obtained by MVSPL are superior to other competing methods with varying noise control parameters σ. For instance, with noise parameter $\sigma = 0.4$, the average test accuracy of MVSPL is 87.84% superior to 85.04%, 84.96%, 87.11% and 85.44% obtained by L_1, L_{EN}, SPL and Ensemble_EN, respectively. In addition, the average test prediction performance of Ensemble_EN performs better than the single-view based methods L_1 and L_{EN} in all cases of Scenario 2.

Table 1 shows the variable selection performance of all the five methods in Scenarios 1 and 2. β-sensitivity and β-specificity are used to evaluate the variable selection performance. It can be obviously seen that our method achieves the best β-sensitivity performance across all cases of simulated experiments. For instance, with noise parameters $\sigma = 0.6$, the average β-sensitivity performance of MVSPL is 91.73% higher than 91.12%, 91.94%, 90.23% and 91.67% obtained by L_1, L_{EN}, SPL and Ensemble_EN, respectively. Moreover, by analyzing more views of data, it can improve the β-sensitive performance and help identify the significant variables. The average β-sensitivity of MVSPL and Ensemble_EN are superior to other single-view analysis methods in most cases. For example, the average β-sensitivity of MVSPL and Ensemble_EN are 91.09% and 90.34% better than 88.18%,

Dataset	No. of Probes	Classes (Class1/Class2)	No. of Classes (Class1/ Class2)	Affymetrix Platform
GSE10072	22284	Normal/Tumor	107 (49/58)	U133A
GSE19188	54675	Normal/Tumor	179 (88/91)	U133 Plus 2.0
GSE19804	54676	Normal/Tumor	120 (60/60)	U133 Plus 2.0
GSE43346	22283	Normal/Tumor	65 (42/23)	U133A

Table 3. Four publicly available lung cancer gene expression datasets used in the real data experiments.

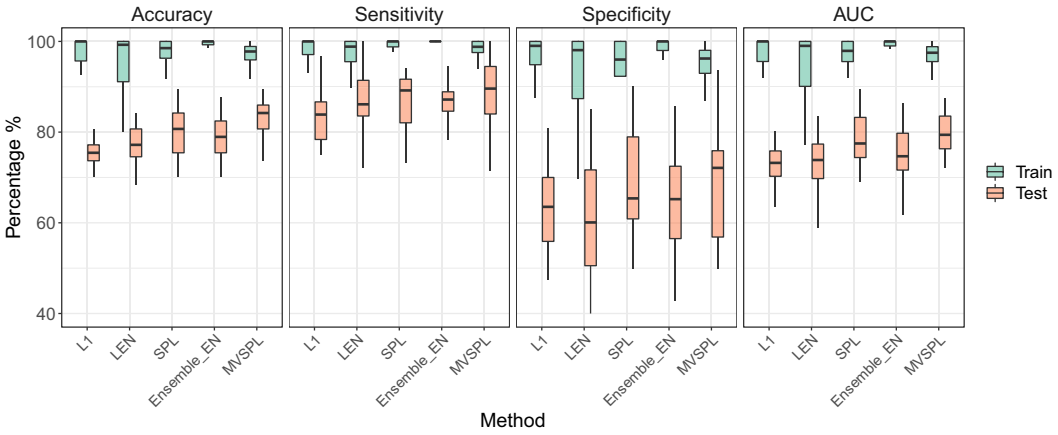

Figure 5. Boxplot diagram of training and test prediction performance for the methods with 30 repetitions in breast cancer dataset.

88.91% and 88.48% obtained by L_1, L_{EN} and SPL with the noise parameter $\sigma = 0.8$. The β-specificity of all the methods is relatively close in different parameters, between 97.0% to 99%.

In Scenario 3, we explored the effect of different sample sizes on the performance of the five methods. As shown in Fig. 4, we can clearly observe that the test accuracy of MVSPL has achieved the optimal results. MVSPL and SPL exhibit better generalization capabilities compared to other methods, especially in high noise case $\sigma = 0.8$. Furthermore, the test accuracy of multi-view based method Ensemble_EN is superior to the single-view based methods L_1 and L_{EN} in Scenario 3.

To further evaluate the performance of the proposed MVSPL method, we designed Scenario 4 in the simulated experiment. The prediction performance of MVSPL in the different number of views is shown in Supplementary Fig. S2. When the number of views increases, the accuracy, sensitivity, specificity and AUC for the test dataset obtained by MVSPL are improved. And we also compare the prediction performance of MVSPL in three views and each of its views. Supplementary Fig. S3 clearly shows that the prediction performance in each single views of MVSPL is worse than that of MVSPL in all views.

To sum up, according to the results of simulated experiments, we can conclude that:

- MVSPL achieves the best generalization ability than the competing methods. The performance of MVSPL outperforms other competing methods with varying correlation parameters and noise parameters.
- By analyzing more views of data, it possible to improve the prediction and variable selection performance. The average performance of MVSPL and Ensemble_EN are superior to the corresponding single-view based methods in most cases.
- When the number of views increases, the prediction performance of MVSPL are improved. This implies that batch effects have an effect for data analysis and more views will contain more comprehensive information.

Real microarray datasets. We curated data from eight publicly available microarray studies, four breast cancer datasets (same platform) and four lung cancer datasets (disparate platform) (Tables 2 and 3). All of these four breast datasets were produced by the same microarray platform HG-U133A. Classification of breast cancer samples aims to distinguish between the sample's estrogen receptor (ER) status (+ve or −ve). Four publicly available lung cancer microarray datasets come from disparate platforms. All these publicly available cancer gene expression datasets can be download from GEO (https://www.ncbi.nlm.nih.gov/geo/).

Analysis of real data. For the real microarray data, two types of experimental designs are used in this work. One type evaluates the performance using a random partition. The other type validates the prediction

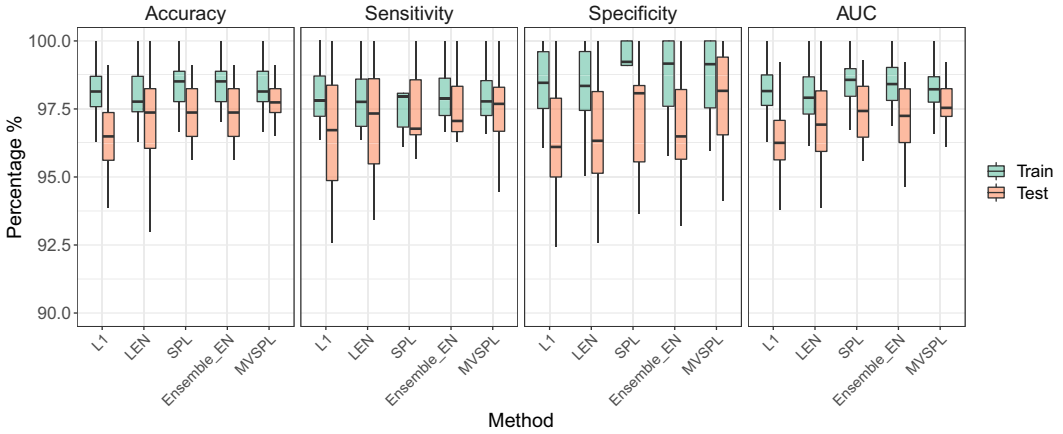

Figure 6. Boxplot diagram of training and test prediction performance for the methods with 30 repetitions in lung cancer dataset.

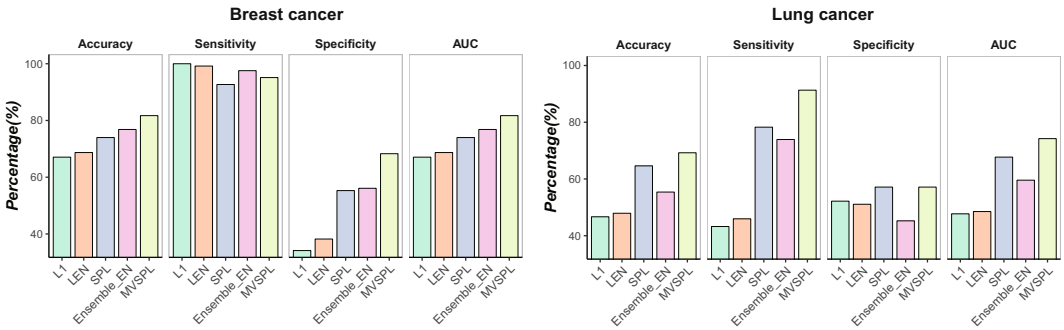

Figure 7. Validation performance comparisons of different integrative analysis methods in the validation datasets of breast cancer and lung cancer studies.

performance on the independent datasets. All publicly available cancer datasets are processed and aggregated in the manner described above (Supplementary Tables S1 and S2). All of publicly available gene expression datasets used in this paper have the class information. Special note, L_1, L_{EN} and SPL achieve the best performance in the view of data by using ComBat_p to eliminate the batch effects. Therefore, these three methods use this view of the aggregated dataset for data analysis in real data analysis. MVSPL and Ensemble_EN analyze two views of data in the real data experiments, which use ComBat_p and ComBat_n to eliminate the batch effects.

Evaluating the performance using a random partition. For the part of evaluating the performance using a random partition, we randomly divide the datasets such that 70% of the datasets become the training samples and the remaining samples become the test samples. The estimation of the optimal regularization parameter λ of the training dataset is obtained by 10-fold cross-validation. We repeat this procedure 30 times and report the average measurement and standard error.

Figures 5 and 6 plot the box plot analysis of training and test prediction performance calculated on breast and lung cancer datasets under 30 repetitions, respectively. As shown in Fig. 5, for the training dataset, all the five methods achieve desirable performance. For instance, the median average training accuracy of all methods have obtained more than 94%. For the test dataset, the proposed MVSPL has the superior performance compared to other competing methods. For example, the median test accuracy of MVSPL is 84.21%, which is obviously better than 75.44%, 77.19%, 80.70% and 76.90% obtained by L_1, L_{EN}, SPL and Ensemble_EN, respectively. Our method achieves the best generalization ability than the competing methods. For lung cancer dataset, as shown in Fig. 6, the training and test prediction performance of all the five methods have reached more than 90%. Our proposed MVSPL method still obtains better classification accuracy, sensitivity, specificity and AUC than other methods. The average number of selected genes for all methods is summarized in Supplementary Table S3.

Validating the classifier on independent dataset. For the part of validating the classifier on independent dataset, the design of the validation process is the same as that of metAnalyzeAll[22]. After pre-processing each dataset individually, all the training datasets and the independent validation dataset are merged in the manner described above. The classifier is trained on the samples from the aggregated training dataset and the optimal regularization

			Ensemble_EN		MVSPL	
L_1	L_{EN}	SPL	View1	View2	View1	View2
SNAPC5	**SNAPC5**	RPL7P25	GNL3LP1	**SNAPC5**	CASP5*	CASP5*
KPNA5	RHCG	**CDK14**	SNAPC5	GNL3LP1	**ALOX15**	GFI1B*
RHCG	**ALOX15**	CCNC	XYLB	XYLB	**SNAPC5**	**ALOX15**
ALOX15	**CDK14**	RHCG	UTRN	APOBEC1	SLC28A2*	**CDK14**
ANXA2P3	KPNA5	ANXA2P3	SMG8	**CDK14**	**CDK14**	UPK3A
CDK14	SMG8	MRM2	ACO1	UTRN	GFI1B*	SLC28A2*
CCNC	AHCYL1	POLR2G	AHCYL1	**RPL7P25**	UPK3A	TNFSF11
PCBP2	**PCBP2**	**GNA13**	**CDK14**	APOO	TNFSF11	RNASE2*
AHCYL1	APOO	UPK3A	APOBEC1	RHCG	CCNC	CCNC
SMG8	CCNC	**ALOX15**	**ALOX15**	AHCYL1	UBE2I*	**SNAPC5**
APOO	**GNA13**	RBBP9	SLC25A31	RIMS2	MPZL2*	**PCBP2**
ANAPC10	ANXA2P3	HIGD1B	SERPINB8	SMG8	**PCBP2**	FGGY*
SRD5A2	UTRN	NUBP2	APOO	ACO1	SETX*	MAT2A*
RPL7P25	MRM2	SMG8	KPNA5	**PCBP2**	NNAT*	**RPL7P25**
MRM2	ANAPC10	**SNAPC5**	RIMS2	SERPINB8	IRGQ*	NNAT*
GNA13	TRIM13	UBA5	**GNA13**	AKTIP	**GNA13**	ALDH1L1*
COX7BP1	ACO1	TNFSF11	AFDN	FA2H	RNASE2*	SENP6
NUBP2	**RPL7P25**	AKTIP	**PCBP2**	LIPC	NUBP2	IRGQ*
TRIM13	RIMS2	WWOX	RHCG	**ALOX15**	RETREG3*	SETX*
UTRN	BANP	**PCBP2**	WWOX	FOXK2	MAT2A*	FOXK2

Table 4. Top 20 genes selected from different integrative analysis methods in breast cancer dataset. [1]The genes with star (*) are the unique gene selected by MVSPL, and the common genes selected by each method are emphasized with bold.

			Ensemble_EN		MVSPL	
L1	LEN	SPL	View1	View2	View1	View2
HTN3	**HTN3**	**HTN3**	**HTN3**	**MYH1**	OR1G1	**HTN3**
MYH1	**MYH1**	**MYH1**	**MYH1**	**HTN3**	**HTN3**	OR1G1
DCC	**DCC**	**DCC**	**DCC**	**DCC**	**MYH1**	**GH2**
RBM15B	**TRBV10-2**	**TRBV10-2**	**TRBV10-2**	RBM15B	**GH2**	**MYH1**
TRBV10-2	**TRPC3**	**GH2**	**TRPC3**	**TRBV10-2**	MASP1	MLNR*
TRPC3	RBM15B	NEUROG1	**GH2**	NEUROG1	**TRBV10-2**	**TRBV10-2**
KLHL21	**GH2**	PITPNA	RBM15B	OR1G1	**ZNF107**	**ZNF107**
TRAM2	NEUROG1	**TRPC3**	NEUROG1	**TRPC3**	**TRPC3**	**DCC**
NEUROG1	TRAM2	OR1G1	PITPNA	**KLHL21**	MLNR*	**PHEX**
GGT5	**GGT5**	RBM15B	TRAM2	EXD3	**GGT5**	IGHE*
GH2	PITPNA	MASP1	**GGT5**	**GGT5**	**KLHL21**	FAM120C*
EXD3	EXD3	TRAM2	EXD3	TRAM2	ZNF254*	KLHL21
TTN	**ZNF107**	**GGT5**	**ZNF107**	CARHSP1	**PHEX**	**GGT5**
ZNF107	**KLHL21**	OR12D3	**KLHL21**	PITPNA	**DCC**	FAF2*
CARHSP1	TTN	EXD3	OR1G1	**GH2**	TMX2	ADAM3A*
PITPNA	OR12D3	**ZNF107**	TTN	TMX2	CARHSP1	BRD7P3*
MFSD11	OR1G1	**PHEX**	OR12D3	**PHEX**	TTN	MFSD11
PHEX	**PHEX**	TTN	**PHEX**	**ZNF107**	MFSD11	**TRPC3**
TMX2	MFSD11	**KLHL21**	MFSD11	MFSD11	IGHE*	AMELX*
LRCH1	CARHSP1	CAMSAP1	CARHSP1	TTN	RPL10L*	MASP1

Table 5. Top 20 genes selected from different integrative analysis methods in lung cancer dataset. [1]The genes with star (*) are the unique gene selected by MVSPL, and the common genes selected by each method are emphasized with bold.

parameter λ is obtained by 10-fold cross-validation. After that, the classifier is tested on the samples from the independent validation dataset.

Figure 7 compares the validation prediction performance of L_1, L_{EN}, SPL, Ensemble_EN and MVSPL in the validation datasets of breast cancer and lung cancer studies. Validating classifiers on the validation dataset, MVSPL consistently outperforms other competing methods in cancer classification problem. As shown in the left hand of Fig. 7, in breast cancer study, the validation accuracy, specificity, and AUC of MVSPL is superior to other

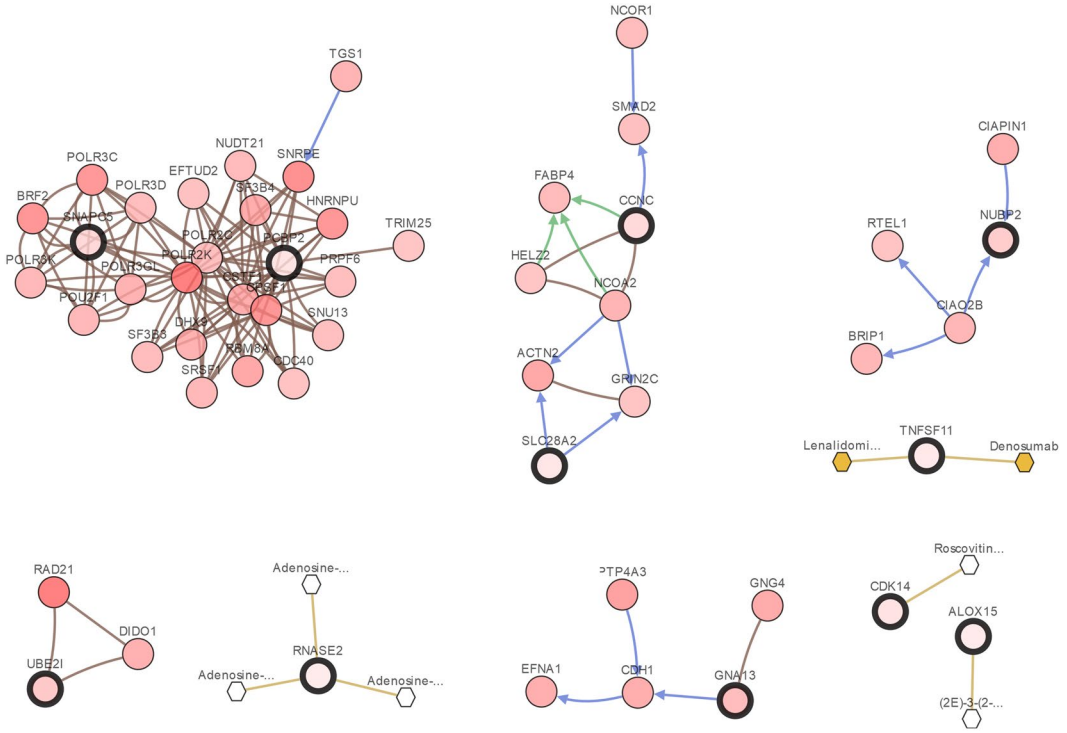

Figure 8. Integrative network view of the genes selected from MVSPL in breast cancer study. The genes corresponding to the selected features are highlighted by a thicker black outline. The rest of the nodes correspond to the genes that are frequently altered and are known to interact with the highlighted genes (based on publicly available interaction data). The nodes are gradient color-coded according to the alteration frequency based on microarray data derived from the TCGA breast cancer dataset via cBioPortal.

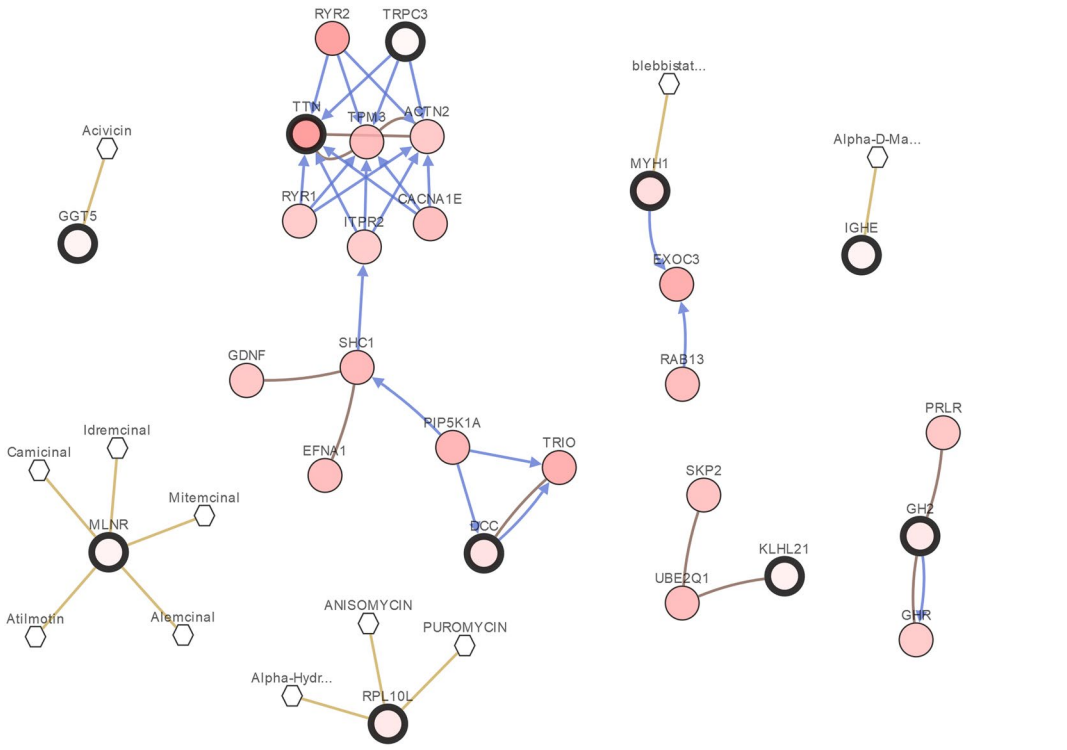

Figure 9. Integrative network view of the genes selected from MVSPL in lung cancer study.

competing methods, except for sensitivity. Specially, MVSPL achieves approximate 10% validation accuracy gain compared with L_1 and L_{EN}. Beyond that, Ensemble_EN with the suboptimal performance. In breast cancer study, multi-view analysis method performs better validation prediction performance than single-view analysis method. For lung cancer study, as shown in the right hand of Fig. 7, the validation prediction performance of the proposed MVSPL method has a significant improvement compared to other methods. For example, the validation sensitivity of MVSPL is 91.30%, which is superior to 43.24%, 45.95%, 78.26% and 73.91% obtained by L_1, L_{EN}, SPL and Ensemble_EN, respectively. The validation prediction performance of SPL is inferior to MVSPL but is obviously superior to L_1, L_{EN} and Ensemble_EN. Moreover, the validation results of Ensemble_EN is outperformed than L_1 and L_{EN}. To summary, by learning from easy to complex samples and interact with multiple views, MVSPL with the best generalization ability than other competing methods. Generally speaking, MVSPL can be successfully applied to the microarray integrative analysis in cancer classification. The average number of selected genes for all methods is summarized in Supplementary Table S4.

For a brief biological analysis of selected genes, we summaries of the 20 top-ranked genes selected by the five integrative analysis methods in two cancer studies, which are shown in Tables 4 and 5, respectively. To make it easier to demonstrate the interplay between the top selected genes from the microarray integrative analysis, we constructed an network of interactions among the genes using the cBioPortal[50,51]. Figure 8 shows the interactive network of the 20 top-ranked genes selected by MVSPL in breast cancer study. The interactive network shows that SNAPC5, PCBP2 and GNA13 are connected to other frequently altered genes from the TCGA breast invasive carcinoma dataset, which are also selected by other competing methods. Moreover, TNFSF11 is targeted by two FDA approved cancer drugs, it is selected only by MVSPL and SPL. For the genes that are only selected by MVSPL, UBE21 is connected to other frequently altered genes and RNASE2 is targeted by three cancer drugs. For lung cancer study, Fig. 9 shows the interactive network of the 20 top-ranked genes obtained by the proposed MVSPL in lung cancer study. Examination of the resulting network, Fig. 9 shows that TRPC3, DCC, MYH1, GH2 and KLHL21 are linked to other frequently altered genes from the TCGA lung adenocarcinoma dataset. MYH1 and GGT5 are targeted by certain cancer drugs. Moreover, MLNR, IGHE and RPL10L are only obtained by MVSPL, these genes are targets for cancer drugs.

In addition, a number of genes selected by the five methods have been reported in the literature. For example, in breast cancer, downregulation of ALOX15 expression has been reported in[52,53]. The upregulated expression of CDK14 promotes tumor cell proliferation, migration and invasion through Wnt/β— catenin signaling pathway in breast cancer[54]. UPK3A is highly expressed in breast cancer[55], which is selected only by MVSPL and SPL. Beyond that, MVSPL selects some other unique genes compared with other methods. Phuong *et al.*[56] confirmed that MAT2A expression in TAM-resistant human breast cancer tissues was higher than that in TAM-responsive cases. Nass *et al.*[57] proposed that NNAT expression determined by immunohistochemistry might therefore become a helpful additional biomarker to identify high-risk breast cancer patients. For lung cancer, Greenman *et al.*[58] reported in 2005 that the role of TTN as a cancer gene is currently a mathematically based prediction and will require direct biological evaluation. And after a few years, Tan H *et al.*[59] said TTN and/or MUC16 were retained in the top 10 for lung cancer, suggesting their tumorigenic relevance to these cancers. MASP1 is over expressed in lung cancer[60]. In this part, we analysis the 20 top-ranked genes selected by the five methods in two cancer studies in gene level. According to the network of interactions among the genes, we find a few numbers of genes are connected to other frequently altered genes from the publicly available datasets and some genes are targeted by certain cancer drugs.

Conclusion

Due to the complexity of gene expression data, there are four major issues constrain the development of microarray technology in clinical applications: high noise, large p & small n problem, batch effects and low reproducibility of significant biomarkers. In this work, we design a novel framework called MVIAm to strive to tackle these issues. MVIAm utilizes different cross-platform normalization methods to minimize the impact of batch effects, keeps as much useful information as possible in the microarray gene expression data. In addition, the aggregated gene expression datasets generated by MVIAm belong to multi-view data. It implies that MVIAm can significantly alleviate the large p & small n problem compared to the existing integrative analysis methods. Therefore, MVIAm can increase the statistical power in identifying the significant biomarkers. To analysis of multi-view gene expression data, we propose a robust learning mechanism called MVSPL to minimize high noise interference. The MVSPL method can improve the generalization performance by learning multi-view data in a meaningful order and improve the prediction performance by the interaction between multiple views. MVSPL actually corresponds to the sum of SPL model under multiple views plus a regularization term. This method implements robust learning regimes in multiple views under the regularization that the robust loss forms in multiple views are closely related. According to the results of simulation and real data experiments, MVSPL has the superior performance compared with L_1, L_{EN}, SPL and Ensemble_EN. Especially in the test and validation dataset, MVSPL shows prominent generalization performance. In a word, MVSPL is a feasible and effective method for variable selection and classification in high dimensional data.

There are some ongoing challenges and promising directions that motivate future work. First, our proposed method conducts variable selection with aggregated microarray data in an "all-in-or-all-out" fashion, that is, a gene identified in all of studies or not identified in any study. However, due to data heterogeneity, there may be some genes are important in some studies while unimportant in others. In the future, we will take this situation into account to improve our model. Second, rapid advances in technology have led to a vast quantity of large-scale molecular omics datasets, it provides a distinct view of the complex biological system. Multi-omics dataset with the same set of samples but several distinct feature sets, which naturally belongs to multi-view data. In the future, we will apply our method to the analysis of multi-omics data. We think the computational analysis of the multi-omics data provides an unprecedented opportunity to deepen our understanding of complex cancer mechanisms. Our proposed method makes integrative analysis more systematic and expands its range of applications.

Data Availability

The code of this paper can be download from https://github.com/must-bio-team/MVIAm.

References

1. Barrett, T. *et al*. NCBI GEO: archive for functional genomics data sets-update. *Nucleic acids research* **41**, D991–D995 (2012).
2. Pepe, M. S. & Feng, Z. Improving biomarker identification with better designs and reporting. *Clinical Chemistry* 1093–1095 (2011).
3. Draghici, S. Statistical intelligence: effective analysis of high-density microarray data. *Drug discovery today* **7**, S55–S63 (2002).
4. Kitchen, R. R. *et al*. Relative impact of key sources of systematic noise in affymetrix and illumina gene-expression microarray experiments. *BMC genomics* **12**, 589 (2011).
5. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M. & Herrera, F. A review of microarray datasets and applied feature selection methods. *Inf. Sci* **282**, 111–135 (2014).
6. Wang, Y., Miller, D. & Clarke, R. Approaches to working in high-dimensional data spaces: gene expression microarrays. *Br. journal cancer* **98**, 1023 (2008).
7. Liang, Y. *et al*. Sparse logistic regression with a $L^{1/2}$ penalty for gene selection in cancer classification. *BMC bioinformatics* **14**, 198 (2013).
8. Yang, Z. Y. *et al*. Robust sparse logistic regression with the $L_q(0 < q < 1)$ regularization for feature selection using gene expression data. *IEEE Access* **6**, 68586–68595 (2018).
9. Larkin, J. E., Frank, B. C., Gavras, H., Sultana, R. & Quackenbush, J. Independence and reproducibility across microarray platforms. *Nat. methods* **2**, 337 (2005).
10. Leek, J. T. *et al*. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.* **11**, 733 (2010).
11. Shen, R., Chinnaiyan, A. M. & Ghosh, D. Pathway analysis reveals functional convergence of gene expression profiles in breast cancer. *BMC medical genomics* **1**, 28 (2008).
12. Tseng, G. C., Ghosh, D. & Feingold, E. Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic acids research* **40**, 3785–3799 (2012).
13. Sørlie, T. *et al*. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proc. national academy sciences* **100**, 8418–8423 (2003).
14. Hamid, J. S. *et al*. Data integration in genetics and genomics: methods and challenges. *Hum. genomics proteomics: HGP* **2009** (2009).
15. Rhodes, D. R. *et al*. Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc. Natl. Acad. Sci.* **101**, 9309–9314 (2004).
16. Choi, J. K., Yu, U., Kim, S. & Yoo, O. J. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics* **19**, i84–i90 (2003).
17. Chang, L.-C., Lin, H.-M., Sibille, E. & Tseng, G. C. Meta-analysis methods for combining multiple expression profiles: comparisons, statistical characterization and an application guideline. *BMC bioinformatics* **14**, 368 (2013).
18. Lusa, L., Gentleman, R. & Ruschhaupt, M. Genemeta: metaanalysis for high throughput experiments. R package version **1** (2006).
19. Parmigiani, G., Garrett, E. S., Anbazhagan, R. & Gabrielson, E. A statistical framework for expression-based molecular classification in cancer. *J. Royal Stat. Soc. Ser. B (Statistical Methodol.)* **64**, 717–736 (2002).
20. Ma, S. & Huang, J. Regularized gene selection in cancer microarray meta-analysis. *BMC bioinformatics* **10**, 1 (2009).
21. Li, Q., Wang, S., Huang, C.-C., Yu, M. & Shao, J. Meta-analysis based variable selection for gene expression data. *Biometrics* **70**, 872–880 (2014).
22. Hughey, J. J. & Butte, A. J. Robust meta-analysis of gene expression using the elastic net. *Nucleic acids research* **43**, e79–e79 (2015).
23. Walsh, C., Hu, P., Batt, J. & Santos, C. Microarray meta-analysis and cross-platform normalization: integrative genomics for robust biomarker discovery. *Microarrays* **4**, 389–406 (2015).
24. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* **8**, 118–127 (2007).
25. Shabalin, A. A., Tjelmeland, H., Fan, C., Perou, C. M. & Nobel, A. B. Merging two gene-expression studies via cross-platform normalization. *Bioinformatics* **24**, 1154–1160 (2008).
26. Giordan, M. A two-stage procedure for the removal of batch effects in microarray studies. *Stat. Biosci.* **6**, 73–84 (2014).
27. Chen, C. *et al*. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS one* **6**, e17238 (2011).
28. Li, Y., Wu, F.-X. & Ngom, A. A review on machine learning principles for multi-view biological data integration. *Briefings bioinformatics* **19**, 325–340 (2016).
29. Li, Y., Yang, M. & Zhang, Z. M. A survey of multi-view representation learning. *IEEE Transactions on Knowl. Data Eng.* (2018).
30. Zhao, J., Xie, X., Xu, X. & Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **38**, 43–54 (2017).
31. Singh, A. *et al*. Diablo: an integrative approach for identifying key molecular drivers from multi-omics assays. *Bioinformatics* (2019).
32. Kumar, M. P., Packer, B. & Koller, D. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 1189–1197 (2010).
33. Shu, J. *et al*. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. *arXiv preprint arXiv*, 1902.07379 (2019).
34. Bengio, Y., Louradour, J., Collobert, R. & Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48 (ACM, 2009).
35. Kumar, M. P., Turki, H., Preston, D. & Koller, D. Learning specific-class segmentation from diverse data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1800–1807 (IEEE, 2011).
36. Tang, K., Ramanathan, V., Fei-Fei, L. & Koller, D. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, 638–646 (2012).
37. Jiang, L., Meng, D., Mitamura, T. & Hauptmann, A. G. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, 547–556 (ACM, 2014).
38. Chai, H., Li, Z.-N., Meng, D.-Y., Xia, L.-Y. & Liang, Y. A new semi-supervised learning model combined with cox and sp-aft models in cancer survival analysis. *Sci. reports* **7**, 13053 (2017).
39. Meng, D., Zhao, Q. & Jiang, L. A theoretical understanding of self-paced learning. *Inf. Sci.* **414**, 319–328 (2017).
40. Irizarry, R. A. *et al*. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4**, 249–264 (2003).
41. Gentleman, R. C. *et al*. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology* **5**, R80 (2004).
42. Friedman, J., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. statistical software* **33**, 1 (2010).
43. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. Ser. B (Methodological)* 267–288 (1996).
44. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *J. Royal Stat. Soc. Ser. B (Statistical Methodol.)* **67**, 301–320 (2005).
45. Günther, O. P. *et al*. A computational pipeline for the development of multi-marker bio-signature panels and ensemble classifiers. *BMC bioinformatics* **13**, 326 (2012).
46. Sohn, I., Kim, J., Jung, S.-H. & Park, C. Gradient lasso for cox proportional hazards model. *Bioinformatics* **25**, 1775–1781 (2009).

47. Baratloo, A., Hosseini, M., Negida, A. & El Ashal, G. Part 1: simple definition and calculation of accuracy, sensitivity and specificity. *Emergency* **3**, 48–49 (2015).
48. Lobo, J. M., Jiménez-Valverde, A. & Real, R. Auc: a misleading measure of the performance of predictive distribution models. *Glob. ecology Biogeogr.* **17**, 145–151 (2008).
49. Zhang, W. *et al*. Molecular pathway identification using biological network-regularized logistic models. *BMC genomics* **14**, S7 (2013).
50. Gao, J. *et al*. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Sci. Signal.* **6**, pl1–pl1 (2013).
51. Cerami, E. *et al*. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data (2012).
52. Jiang, W. G., Watkins, G., Douglas-Jones, A. & Mansel, R. E. Reduction of isoforms of 15-lipoxygenase (15-lox)-1 and 15-lox-2 in human breast cancer. *Prostaglandins, Leukot. Essent. Fat. Acids* **74**, 235–245 (2006).
53. Ho, C. F.-Y. *et al*. Expression of dha-metabolizing enzyme alox15 is regulated by selective histone acetylation in neuroblastoma cells. *Neurochem. research* **43**, 540–555 (2018).
54. Gu, X. *et al*. Upregulated pftk1 promotes tumor cell proliferation, migration, and invasion in breast cancer. *Med. Oncol.* **32**, 195 (2015).
55. Network, C. G. A. R. *et al*. Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature* **507**, 315 (2014).
56. Phuong, N. T. T. *et al*. Induction of methionine adenosyltransferase 2a in tamoxifen-resistant breast cancer cells. *Oncotarget* **7**, 13902 (2016).
57. Nass, N. *et al*. High neuronatin (nnat) expression is associated with poor outcome in breast cancer. *Virchows Arch.* **471**, 23–30 (2017).
58. Greenman, C. *et al*. Patterns of somatic mutation in human cancer genomes. *Nature* **446**, 153 (2007).
59. Tan, H., Bao, J. & Zhou, X. Genome-wide mutational spectra analysis reveals significant cancer-specific heterogeneity. *Sci. reports* **5**, 12566 (2015).
60. Kang, J. U., Koo, S. H., Kwon, K. C., Park, J. W. & Kim, J. M. Identification of novel candidate target genes, including ephb3, masp1 and sst at 3q26. 2-q29 in squamous cell carcinoma of the lung. *BMC cancer* **9**, 237 (2009).

Acknowledgements

This work is partially supported by the Chinese Ministry of Education's Tian Cheng Hui Zhi Innovation and Education Improvement Funds (Grant No. 2018A01014), the Macau Science and Technology Develop Funds (Grant No. 0055/2018/A2) of Macao SAR of China and China NSFC project under contract 61661166011.

Author Contributions

Z.Y.Y., J.S. and Y.L. proposed the Novel MVIAm integrative framework and proposed multi-view self-paced learning approach, designed the algorithm, wrote the code and manuscript, X.Y.L., H.Z. and Y.Q.R. provided the real data and analysis the information of biology, Z.B.X. provided the technical support. All authors reviewed the manuscript.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-49967-4.

Competing Interests: The authors declare no competing interests.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.