PLOS ONE

RESEARCH ARTICLE

Classification and prediction for multi-cancer data with ultrahigh-dimensional gene expressions

Li-Pang Chen *

Department of Statistics, National Chengchi University, Taipei, Taiwan, ROC

* lchen723@nccu.edu.tw

OPEN ACCESS

Citation: Chen L-P (2022) Classification and prediction for multi-cancer data with ultrahigh-dimensional gene expressions. PLoS ONE 17(9): e0274440. https://doi.org/10.1371/journal.pone.0274440

Editor: Enrique Hernandez-Lemus, Instituto Nacional de Medicina Genomica, MEXICO

Received: September 27, 2021

Accepted: August 28, 2022

Published: September 15, 2022

Data Availability Statement: All relevant data are within the paper and its Supporting information files.

Funding: This research is supported by Ministry of Science and Technology with grant ID 110-2118-M-004 -006 -MY2. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

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

Abstract

Analysis of gene expression data is an attractive topic in the field of bioinformatics, and a typical application is to classify and predict individuals' diseases or tumors by treating gene expression values as predictors. A primary challenge of this study comes from ultrahigh-dimensionality, which makes that (i) many predictors in the dataset might be non-informative, (ii) pairwise dependence structures possibly exist among high-dimensional predictors, yielding the network structure. While many supervised learning methods have been developed, it is expected that the prediction performance would be affected if impacts of ultrahigh-dimensionality were not carefully addressed. In this paper, we propose a new statistical learning algorithm to deal with multi-classification subject to ultrahigh-dimensional gene expressions. In the proposed algorithm, we employ the model-free feature screening method to retain informative gene expression values from ultrahigh-dimensional data, and then construct predictive models with network structures of selected gene expression accommodated. Different from existing supervised learning methods that build predictive models based on entire dataset, our approach is able to identify informative predictors and dependence structures for gene expression. Throughout analysis of a real dataset, we find that the proposed algorithm gives precise classification as well as accurate prediction, and outperforms some commonly used supervised learning methods.

1 Introduction

Analysis of gene expression data is an important topic in bioinformatics. A large body of research and relevant developments have been explored in recent years. One of important branches of gene expression data analysis is to take gene expression values as predictors to classify and predict tumors to possible cancers. A motivated example in this paper is the GCM dataset, which contains 16,063 gene expression values and 14 human cancers among 198 tumor samples. The goal of this study is to take gene expression values as the predictors, and use them to classify tumor samples to their corresponding cancers. In this dataset, a key feature is ultrahigh-dimensional predictors in the sense that the dimension of predictors (number of gene expression values) is extremely greater than the sample size (tumor samples). This feature

further induces some challenges, including (a) pairwise interactions among gene expressions and (b) existence of non-informative gene expressions, that affect the performance of classification and the accuracy of prediction.

To address classification and prediction for biomedical research, many supervised learning methods have been developed and have been widely applied in machine learning frameworks. With the ignorance of pairwise interactions and existence of non-informative predictors induced by ultrahigh-dimensional predictors, [1] proposed the integration of several heterogeneous cancer series, and performed a multi-class classification. [2] studied multicategory support vector machine (SVM) for the classification of multiple cancer. [3] presented comprehensive discussions of SVM methods. [4] applied SVM ensembers to analyze breast cancer prediction. [5] discussed linear discrimination analysis (LDA) and its application in the microarray. [6] discussed the multi-class analysis by generalized sparse linear discriminant analysis. The detailed and fundamental discussions of those methods can be found in [7, 8], and were reviewed by [9] as well. In recent years, deep learning approaches, such as convolutional neural network (e.g., [10]) or natural language processing (e.g., [11]), have been developed to deal with multicalssification. More applications can be found in some monographs, such as [12–14].

To characterize pairwise interactions among gene expressions, which usually refers to the *network dependence* among gene expressions, we employ *graphical models* that are powerful methods in describing the dependence structure of variables. A general introduction of graphical models can be found in [7] (Chapter 17). In the past literature, graphical models have been used to deal with the classification problem. For example, [15] proposed the network-based support vector machine for the classification of microarray samples for binary classification. [16] discussed the identification of rheumatoid arthritis-related genes by using a network-based support vector machine. [17] proposed network linear discriminant analysis. [18] proposed the nearest neighbor network. Most existing methods focused on binary responses and restricted the predictors to follow the normal distribution because of explorations of the precision matrix. Furthermore, it is intuitive to understand that the network structure of variables in different classes may not be exactly equal to each other. To address this issue, [19, 20] explored SVM and logistic regressions with heterogeneous network structures accommodated, respectively. More recently, [21, 22] developed multiclass discriminant analysis with network structures accommodated. From the perspectives of Bayesian approaches, several methods were also investigated with the network structure incorporated, including [23, 24].

To address non-informative gene expression values in ultrahigh-dimensional data, variable selection or dimension reduction are perhaps commonly used strategies in the past literature. For example, [25] applied unsupervised feature extraction, such as principal component analysis, tensor decomposition, and kernel tensor decomposition, to select potentially important genes. [26] adopted SIS method to do feature screening for gene expressions and combined Nottingham Prognostic Index with a hybrid signature accommodated. With the combination of supervised learning, [27] proposed the penalized method for SVM. [28, 29] explored variable selection based on LDA. Those methods mainly handled the setting that the dimension is smaller than the sample size, however, it is unknown whether those methods are able to deal with the case that the dimension of predictors is much higher than the sample size.

From the two challenges and developments described above, we note that most existing methods deal with either network structure or variable selection but not both. It motivates us to propose a strategy to *simultaneously* retain important predictors and construct the network structure of predictors when doing classification. Our strategy is outlined in Fig 1. Roughly speaking,

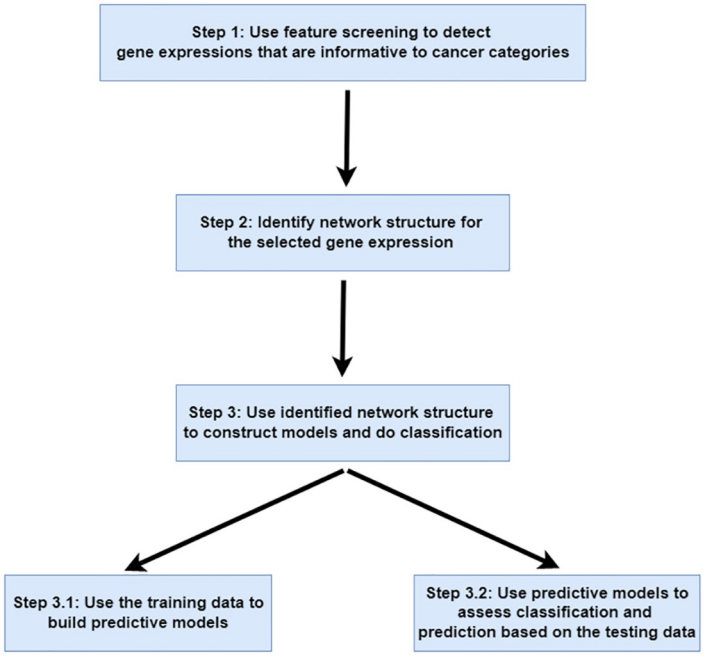

Fig 1. Summary of key steps for the proposed classification method via ultrahigh-dimensional gene expressions.

https://doi.org/10.1371/journal.pone.0274440.g001

(i) to deal with ultrahigh-dimensional predictors where the dimension of predictors is extremely greater than the sample size, we adopt feature screening techniques to retain predictors that are informative to the response;

(ii) to detect network structures of predictors, we employ exponential family graphical models to detect network structure of the selected predictors under the whole dataset or different classes;

(iii) use the results in (i) and (ii) to develop network-based classification models to examine class separation and make the prediction for tumor samples.

There are several contributions in the proposed method. First, unlike existing methods that may specify a model when doing feature screening, our feature screening procedure is model-free and does not need to specify the model formulation. Second, although there exist methods handling network structures in classification, they assume a common network structure for predictors of all subjects without taking into account of possible heterogeneity for different classes. Instead, the proposed method is able to construct predictive models with possibly class-dependent network structures of predictors taken into account. Finally, the proposed method is able to handle multi-class labels with the accommodation of network structures in predictors, which is different from existing methods that either handle multiclassification but not use the information of network structure, or simply accommodate network structure to deal with binary classification.

The remainder is organized as following. In Section 2, we introduce a motivated real dataset and its data structure. In addition, we define the relevant mathematical notation. In Section 3, we give detailed presentation for each step in Fig 1. In Section 4, we implement the proposed method to analyze a real dataset and compare the proposed method with its competitors. A general discussion is presented in Section 5.

Table 1. Sample sizes for each cancer. The first row with $\mathcal{T}$ contains sample sizes of the training data in cancer labels; the second row with $\mathcal{V}$ contains sample sizes of the testing data in cancer labels; the last row with "Total" contains sample sizes of the whole data in cancer labels.

	BR	PR	LE	CO	LU	BL	CNS	UT	LY	RE	PA	OV	ME	ML
$\mathcal{T}$	8	8	8	8	16	8	8	8	24	8	8	8	8	16
$\mathcal{V}$	3	4	2	4	4	4	2	3	6	4	3	6	3	6
Total	11	12	10	12	20	12	10	11	30	12	11	14	11	16

https://doi.org/10.1371/journal.pone.0274440.t001

2 Data structure with multi-class responses

In this section, we first introduce a motivated dataset outlined in Section 1. After that, we define mathematical notation to describe the data structure with multi-class responses.

2.1 Description of motivated dataset

The data presented in the following are the GCM dataset collected by [30]. This dataset contains 16,063 gene expression values and 198 tumor samples, including 144 training samples (denoted as $\mathcal{T}$) and 54 testing samples (denoted as $\mathcal{V}$). In addition, 14 common human cancers, including Breast (BR), Prostate (PR), Lung (LU), Colorectal (CO), Lymphoma (LY), Bladder (BL), Melanoma (ML), Uterus (UT), Leukemia (LE), Renal (RE), Pancreas (PA), Ovarym (OV), Mesothelioma (ME) and CNS cancers, are included in the dataset. The sample sizes of each cancer are summarized in Table 1. Our main goal is to classify tumor samples into different categories of cancer according to gene expression values of the samples, which are treated as predictors.

Even though this dataset is no need to pre-processing due to complete observations without missing value, and some of its features having been well analyzed by [30], still, the dataset can be further investigated in two aspects. First of all, we propose to note the issue of high-dimentionality of the data, which usually implies the existence of irrelevant variables, i.e., not every gene expression is dependent upon the response. Therefore, to ensure the accuracy of prediction, it is necessary to exclude irrelevant variables. As a result, it is crucial to select gene expressions that are informative in terms of responses. Secondly, as discussed in [31, 32], complex dependence structures may exist among high-dimensional gene expressions. Therefore, to increase the accuracy of predictions, it is necessary to incorporate the network structure of gene expressions into the classification procedure.

2.2 Notation

In this subsection, we define mathematical notation to describe the data in order to develop the method.

Suppose the data of n subjects come from I classes, where I is a fixed integer greater than 2 and the classes are nominal. Let n_i be the class size in class i with $i = 1, \cdots, I$, and hence $n = \sum_{i=1}^{I} n_i$. Let $\mathbf{Y}$ denote the n-dimensional vector of response with the jth component being $Y_j = i$, which reflects the class membership that the jth subject is in the ith class for $i = 1, \cdots, I$ and $j = 1, \cdots, n$.

Let $p > 1$ denote the dimension of predictors for each subject. Define $\mathbf{X} = [X_{j,l}]$ as the $n \times p$ matrix of predictors for $j = 1, \cdots, n$ and $l = 1, \cdots, p$, where the component $X_{j,l}$ represents the lth predictor for the jth subject. Furthermore, let $X_{j\bullet} = (X_{j,1}, \cdots, X_{j,p})^\top$ denote the p-dimensional predictor vector for the jth subject in the jth row of $\mathbf{X}$ and let $X_{\bullet k} = (X_{1,k}, \cdots, X_{n,k})^\top$ represent the n-dimensional vector of the kth predictor in the kth column of $\mathbf{X}$. In this paper, we

consider a setting that the dimension of the predictors p is ultrahigher than the sample size n, i.e., $p = \exp\{O(n^r)\}$ for some constant $r > 0$ (e.g., [33]).

Without loss of generality, the $\{X_{j\bullet}, Y_j\}$ are treated as independent and identically distributed (i.i.d.) for $j = 1, \cdots, n$. We let lower case letters represent realized values for the corresponding random variables.

The objective of the study is to build models to predict the class label for a new subject with observation $\tilde{X}$.

3 Proposed method

In this section, we present detailed estimation procedure for each step as shown in Fig 1.

3.1 Feature screening via rank-based correlation coefficient

Let

$$\mathcal{I} = \{k : X_{\bullet k} \text{ is dependent on } Y \in \{1, 2, \cdots, I\}\}$$

denote the *true active set* which contains all relevant predictors for the response Y with $q = |\mathcal{I}|$ and $q < n$, and $\mathcal{I}^c$ is the complement of $\mathcal{I}$ that contains all irrelevant predictors for the response Y. Basically, the goal of Step 1 in Fig 1 is to estimate the active set $\mathcal{I}$. When $\mathcal{I}$ is determined, then the associated vector of predictors $X_{\mathcal{I}} = \{X_{\bullet k} : k \in \mathcal{I}\}$ contains important information in terms of the response, and its dimension is smaller than the sample size n. Thus, $X_{\mathcal{I}}$ can be adopted to the subsequent analysis.

The remaining concern is to obtain the estimated active set. Following the spirit of [33], we employ the technique of feature screening, whose idea is to take the correlation of the response and the predictors as a signal, and retain the important predictors with large values of signals. We propose to take the rank-based correlation coefficient as the signal. Specifically, for the kth predictor $X_{\bullet k}$, the rank-based correlation coefficient between $X_{\bullet k}$ and Y is given by (e.g., [34, 35])

$$\omega_k \triangleq \xi(X_{\bullet k}, Y) = \frac{\int \mathrm{var}[E\{\mathbb{I}(Y \geq t)|X_{\bullet k}\}]d\mu(t)}{\int \mathrm{var}\{\mathbb{I}(Y \geq t)\}d\mu(t)}, \tag{1}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and $\mu(\cdot)$ is the law of Y. It can be shown that ω_k is in an interval [0, 1], and a higher value of ω_k indicates a stronger correlation between Y and $X_{\bullet k}$. Therefore, (1) can be regarded as similar to the classical coefficients such as Pearson's correlation.

To implement this idea, we estimate (1) using the sample data. For $j = 1, \cdots, n$, denote $Y_{(j)}$ as the rearranged response according to the sort of the kth predictors $X_{\bullet k}$, i.e., $(X_{(1),k}, Y_{(1)}), \cdots, (X_{(n),k}, Y_{(n)})$ with $X_{(1),k} \leq X_{(2),k} \leq \cdots \leq X_{(n),k}$ and $X_{(j),k}$ being the jth sorted predictor in $X_{\bullet k}$. The corresponding estimator of ω_k is given by [34]:

$$\hat{\omega}_k \triangleq \hat{\xi}(X_{\bullet k}, Y) = 1 - \frac{n\sum_{j=1}^{n-1}|r_{j+1} - r_j|}{2\sum_{j=1}^{n}\ell_j(n - \ell_j)}, \tag{2}$$

where, for $j = 1, \cdots, n$, $\ell_j \triangleq \#\ \{l : Y_{(l)} \geq Y_{(j)}\}$, $r_j \triangleq \#\ \{l : Y_{(l)} \leq Y_{(j)}\}$, and $\#\ \mathcal{A}$ represents the number of elements in a set $\mathcal{A}$. In applications, one can use the R package `XICOR` to compute (2).

Therefore, the estimated active set based on (2) is given by

$$\hat{\mathcal{I}} = \{k : \hat{\omega}_k \geq cn^{-\kappa} \text{ for } k = 1, \cdots, p\}, \tag{3}$$

where c and $\kappa \in (0, 1/2)$ are prespecified threshold values. In applications, one can specify c and κ such that variables with the first $\left[\frac{n}{\log n}\right]$ largest values of $\hat{\omega}_k$ can be retained, where $[\cdot]$ represents the ceiling function (e.g., [33, 35, 36]).

Different from the conventional feature screening method (e.g., [33]), the main advantage of (3) is *model-free feature screening* because it does not impose model formulation, and thus, (3) is able to detect predictors that may have nonlinear relationship with the response Y. Theoretically, by the similar derivations of [35], the *sure screening property* of (3) can be justified. That is, $P(\mathcal{I} \subseteq \hat{\mathcal{I}}) \to 1$ as $n \to \infty$, which ensures that the estimated active set contains truly informative predictors that are dependent on the response with a probability approaching one. Moreover, while there are several methods to deal with feature screening, as examined by [35], (2) generally outperforms other existing approaches and is able to handle oscillatory trajectory between the response and predictors.

When the active set is determined, we then let $X_{j,\hat{\mathcal{I}}} = \{X_{j,k} : k \in \hat{\mathcal{I}}\}$ denote the vector containing all the active predictors for the jth subject, and denote $x_{j,\hat{\mathcal{I}}}$ as the realization values of $X_{j,\hat{\mathcal{I}}}$.

3.2 The expressions of graphical structure

Since the estimated active set $\hat{\mathcal{I}}$ is identified, we now explore the network structure of selected gene expressions in $\hat{\mathcal{I}}$ for Step 2 in Fig 1. *Graphical models* are commonly used strategies to achieve this goal.

The graph is expressed as $G = (V, E)$, where V is the set of the vertices and $E \subset V \times V$ is the set of the edges. In our case, $V \triangleq \hat{\mathcal{I}}$ is treated as selected predictors with $\tilde{q} = |V|$ and E is regarded as pairwise dependence of any two selected predictors. In graphical model frameworks, we start by formulating the distribution function of selected predictors. In this article, we consider exponential family graphical models because it generalizes the commonly used models. The formulation is given by

$$\begin{aligned} P(X_{j,\hat{\mathcal{I}}}; \beta, \Theta) \quad = \quad & \exp\left\{\sum_{r \in V} \beta_r B(X_{j,r}) + \sum_{(s,t) \in E} \theta_{st} B(X_{j,s}) B(X_{j,t}) \right. \\ & \left. + \sum_{r \in V} C(X_{j,r}) - A(\beta, \Theta)\right\}, \end{aligned} \tag{4}$$

where $\beta = (\beta_1, \cdots \beta_{\tilde{q}})^\top$ is the $\tilde{q}$-dimensional parameter vector, $\Theta = [\theta_{st}]$ is a $\tilde{q} \times \tilde{q}$ symmetric matrix, $B(\cdot)$ and $C(\cdot)$ are given functions that reflect the distribution of $X_{\hat{\mathcal{I}}}$ (e.g., [20, 37]), and the function $A(\beta, \Theta)$ is normalizing constant which ensures (4) to be integrated as 1.

Without loss of general interest, we take $B(X_{j,r})$ as the linear function $B(X_{j,r}) = X_{j,r}$ for $r \in V$. In addition, in the graphical model theory, the main interest is the estimation of θ_{st} because of its interpretation that $X_{j,s}$ and $X_{j,t}$ are conditionally dependent if $\theta_{st} \neq 0$. Therefore, to focus on presenting the estimation of θ_{st}, we drop the main effect term, and consider the following

graphical model

$$P(X_{j,\tilde{\mathcal{I}}};\Theta)=\exp\left\{\sum_{(s,t)\in E}\theta_{st}X_{j,s}X_{j,t}+\sum_{r\in V}C(X_{j,r})-A(\Theta)\right\}, \tag{5}$$

where the function $A(\Theta)$ is normalization constant which makes (5) be integrated as 1.

For the estimation method for Θ, one of the famous methods is the conditional inference [38]. Without loss of generality, we consider the vertex s, and define the *neighbourhood set*

$$\mathcal{N}(s)=\{t\in V:(s,t)\in E\}, \tag{6}$$

which collect vertexes that are dependent on the vertex s. To estimate the neighbourhood set of s, it suffices to study the inference of $X_{j,s}|X_{j,V\setminus\{s\}}$, where $X_{j,V\setminus\{s\}}=(X_{j,1},\cdots,X_{j,s-1},X_{j,s+1},\cdots,X_{j,\tilde{q}})$. Let $\theta_s=(\theta_{s1},\cdots,\theta_{s(s-1)},\theta_{s(s+1)},\cdots,\theta_{s\tilde{q}})$ denote the $(\tilde{q}-1)$-dimensional vector of parameters that is associated with $X_{j,V\setminus\{s\}}$. By some algebra, we have

$$P(X_{j,s}|X_{j,V\setminus\{s\}};\theta_s)\quad\propto\quad\exp\left\{X_{j,s}\left(\sum_{t\in V\setminus\{s\}}\theta_{st}X_{j,t}\right)+C(X_{j,s})-D\left(\sum_{t\in V\setminus\{s\}}\theta_{st}X_{j,t}\right)\right\}, \tag{7}$$

where $D(\cdot)$ is a normalization constant ensuring that the integration of (7) is equal to 1. Then the estimator of θ_s, denoted as $\hat{\theta}_s$, is given by

$$\hat{\theta}_s=\underset{\theta_s}{\operatorname{argmin}}\{\ell(\theta_s)+\lambda\|\theta_s\|_1\}, \tag{8}$$

where

$$\ell(\theta_s)=\frac{1}{n}\sum_{i=1}^{n}\left\{-X_{i,s}\left(\sum_{t\in V\setminus\{s\}}\theta_{st}X_{i,t}\right)+D\left(\sum_{t\in V\setminus\{s\}}\theta_{st}X_{i,t}\right)\right\},$$

$\|\cdot\|_1$ is the L_1-norm and λ is the tuning parameter.

In the penalization problem for selecting the variables, estimating the tuning parameter is also a crucial issue. In this paper, we employ the BIC approach (e.g., [39]) to select the tuning parameter λ. To emphasize the dependence on the tuning parameter, we let $\hat{\theta}_s(\lambda)$ denote the estimator obtained from (8). Define

$$\mathrm{BIC}(\lambda)=2n\ell(\hat{\theta}_s(\lambda))+\log(n)\times\mathrm{df}\{\hat{\theta}_s(\lambda)\}, \tag{9}$$

where $\mathrm{df}\{\hat{\theta}_s(\lambda)\}$ represents the number of non-zero elements in $\hat{\theta}_s(\lambda)$ for a given λ. The optimal tuning parameter λ, denoted by $\hat{\lambda}$, is determined by minimizing (9) within suitable ranges of λ. As a result, the estimator of θ_s is determined by $\hat{\theta}_s=\hat{\theta}_s(\hat{\lambda})$.

Finally, the estimated neighbourhood set is given by

$$\hat{\mathcal{N}}(s)=\{t\in V:\hat{\theta}_{st}\neq 0\}. \tag{10}$$

Note that θ_{st} is equal to θ_{ts} since Θ is a symmetric matrix. However, the estimators $\hat{\theta}_{st}$ and $\hat{\theta}_{ts}$ are not equal. To overcome this problem, we apply the AND rule [38], which indicates that the final estimators of $\hat{\theta}_{st}$ and $\hat{\theta}_{ts}$ are determined by their maximum if both $\hat{\theta}_{st}$ and $\hat{\theta}_{ts}$ are nonzero; $\hat{\theta}_{st}$ and $\hat{\theta}_{ts}$ are set to be zero if one of them is zero. Moreover, the estimated set of edges is given

by

$$\hat{E} = \{(s,t) : s \in \hat{\mathcal{N}}(t) \text{ and } t \in \hat{\mathcal{N}}(s)\}. \tag{11}$$

After deriving the estimated set of edges, a crucial question is the relationship of $\hat{E}$ and E. To answer this question, we present the following theorem, which gives an important result for the estimated graph.

Theorem 3.1 (*Network Recovery*)

Suppose E is the set of edges, and let $\hat{E}$ *be the estimated set of edges. Under some regular conditions in* [38], *we have that as* $n \to \infty$,

$$P(\hat{E} = E) \to 1. \tag{12}$$

This result and regular conditions are similar to Section 2.2 in [40] and Theorem 5 (b) in [37]. Theorem 3.1 tells us that based on the mild conditions, the estimated network structure can be recovered to the true network structure.

3.3 Multinomial logistic regression with homogeneous network structure in predictors

After obtaining the estimated network structure based on informative predictors, we wish to use such a network structure to examine the classification for different cancers, as demonstrated in Step 3 of Fig 1. Therefore, to incorporate the network structures of the predictors into a prediction model, we present two methods which can be readily implemented using the R package `glm` for fitting a logistic regression model.

In the first method, called the *multinomial logistic regression with homogeneous network structure in predictors* (MLR-HomoNet), we consider the case where the subjects in different classes share a common network structure in the predictors. To build a prediction model, we make use of the development of the logistic model with multiclass responses ([41], Section 6.1; [42], Section 7.1).

We first identify the pairwise dependence of the predictors using the measurements of all the subjects without distinguishing their class label. Let $\hat{\theta}_{st}$ be the estimate for θ_{st} obtained for (8) by using all the predictor measurements of $\{X_{j,\hat{\mathcal{I}}} : j = 1, \cdots, n\}$, and let $\hat{E} = \{(s,t) : \hat{\theta}_{st} \neq 0\}$ denote the resulting estimated set of edges.

Next, for $i = 1, \cdots, I$ and $j = 1, \cdots, n$, we let

$$p_i(x_{j,\hat{\mathcal{I}}}) \triangleq P(Y_j = i | X_{j,\hat{\mathcal{I}}} = x_{j,\hat{\mathcal{I}}})$$

be the conditional probability of $Y_j = i$ given $X_{j,\hat{\mathcal{I}}} = x_{j,\hat{\mathcal{I}}}$. Consider the parametric multinomial logistic model

$$p_i(x_{j,\hat{\mathcal{I}}}) \triangleq p_i(x_{j,\hat{\mathcal{I}}}; \alpha) = \frac{\exp\left(\alpha_{i0} + \sum_{(s,t)\in\hat{E}} \alpha_{i,st} x_{j,s} x_{j,t}\right)}{1 + \sum_{l=1}^{I-1} \exp\left(\alpha_{l0} + \sum_{(s,t)\in\hat{E}} \alpha_{l,st} x_{j,s} x_{j,t}\right)} \tag{13}$$

for $i = 1, 2, \cdots, I-1$, where $\alpha = (\alpha_1^\top, \cdots, \alpha_{I-1}^\top)^\top$ is the vector of parameters with vectors $\alpha_i \triangleq (\alpha_{i0}, \alpha_{i\bullet}^\top)^\top$ and $\alpha_{i\bullet} = (\alpha_{i,st} : (s,t) \in \hat{E})^\top$ reflecting parameters for class i, and the constraint $\sum_{i=1}^{I} p_i(x_{j,\hat{\mathcal{I}}}) = 1$ is imposed for every $j = 1, \cdots, n$.

For subject $j = 1, \cdots, n$, we let $Y_{ij}^* = 1$ if subject j is in class i and $Y_{ij}^* = 0$ otherwise, and hence, $\sum_{i=1}^{I} Y_{ij}^* = 1$ for every j. Let y_{ij}^* denote a realized value of Y_{ij}^*. For $i = 1, \cdots, I$ and $j = 1, \cdots, n$, the log-likelihood function is given by ([42], p.273)

$$L(\alpha) = \sum_{i=1}^{I}\sum_{j=1}^{n} y_{ij}^* \log\{p_i(x_{j,\hat{\mathcal{I}}}; \alpha)\}. \tag{14}$$

The estimator of α, denoted $\hat{\alpha}$, can be derived by maximizing (14). In applications, since $\hat{\alpha}$ has no closed form, we usually implement the Newton-Raphson algorithm to (14) and obtain the resulting estimator. Therefore, for the realization $x_{j,\hat{\mathcal{I}}}$ of the q-dimensional vector $X_{j,\hat{\mathcal{I}}}$, $p_i(x_{j,\hat{\mathcal{I}}})$ is estimated as

$$\hat{p}_i(x_{j,\hat{\mathcal{I}}}) \triangleq p_i(x_{j,\hat{\mathcal{I}}}; \hat{\alpha}) = \frac{\exp\left(\hat{\alpha}_{i0} + \sum_{(s,t)\in\hat{E}} \hat{\alpha}_{i,st} x_{j,s} x_{j,t}\right)}{1 + \sum_{l=1}^{I-1} \exp\left(\hat{\alpha}_{l0} + \sum_{(s,t)\in\hat{E}} \hat{\alpha}_{l,st} x_{j,s} x_{j,t}\right)} \tag{15}$$

for $i = 1, \cdots, I-1$, and $p_I(x_{j,\hat{\mathcal{I}}})$ is estimated as

$$\hat{p}_I(x_{j,\hat{\mathcal{I}}}) = 1 - \sum_{i=1}^{I-1} \hat{p}_i(x_{j,\hat{\mathcal{I}}}). \tag{16}$$

Finally, to predict the class label for a new subject with a selected $\tilde{q}$-dimensional predictor instance $\tilde{x}$, we first calculate the right-hand side of (15) and (16), and let $\tilde{\hat{p}}_1, \cdots, \tilde{\hat{p}}_I$ denote the corresponding values. Let i^* denote the index which corresponds to the largest value of $\{\tilde{\hat{p}}_1, \cdots, \tilde{\hat{p}}_I\}$, i.e., $i^* = \underset{i\in\{1,\cdots,I\}}{\operatorname{argmax}}\ \tilde{\hat{p}}_i$. Then the class label for this new subject is predicted as i^*.

To the end, we summarize key steps in Sections 3.1–3.3 in Algorithm 1.

Algorithm 1: MLR-HomoNet

Under the training data $\mathcal{T}$;

Step 1: Determine informative predictors

Apply (2) to do feature screening and retain $\left[\frac{n}{\log n}\right]$ predictors among p-dimensional predictors. A set of selected predictors is given by (3).

Step 2: Determine the network structure of predictors

Based on selected predictors in $\hat{\mathcal{I}}$, use (8) to determine pairwise dependence structure and obtain (11). The resulting network structure is formed by $\hat{E}$.

Step 3: Construct the predictive model

Given a initial value $\alpha^{(0)}$, then perform the following Newton-Raphson algorithm;

for *step t with* $t = 1, 2, \cdots, T$, *say* $T = 1000$ **do**

Step 3.1: calculate the score function evaluated at the tth iterated value:

$$S(\alpha^{(t)}) \triangleq \left(\frac{\partial L(\alpha)}{\partial \alpha_1}\bigg|_{\alpha=\alpha^{(t)}}, \cdots, \frac{\partial L(\alpha)}{\partial \alpha_{I-1}}\bigg|_{\alpha=\alpha^{(t)}}\right)^{\top}$$

with

$$\left.\frac{\partial L(\alpha)}{\partial \alpha_i}\right|_{\alpha=\alpha^{(t)}} = \sum_{j=1}^{n}\{y_{ij}\left(x_{j,\hat{\mathcal{I}}} - p_i(x_{j,\hat{\mathcal{I}}};\alpha^{(t)})\right) - y_{Ij}p_i(x_{j,\hat{\mathcal{I}}};\alpha^{(t)})\}.$$

Step 3.2: calculate the Henssian matrix evaluated at the tth iterated value:

$$H(\alpha^{(t)}) \triangleq \text{diag}\left(\left.\frac{\partial^2 L(\alpha)}{\partial \alpha_1 \partial \alpha_1^\top}\right|_{\alpha=\alpha^{(t)}}, \cdots, \left.\frac{\partial^2 L(\alpha)}{\partial \alpha_{I-1} \partial \alpha_{I-1}^\top}\right|_{\alpha=\alpha^{(t)}}\right)$$

with

$$\left.\frac{\partial^2 L(\alpha)}{\partial \alpha_i \partial \alpha_i^\top}\right|_{\alpha=\alpha^{(t)}} = -\sum_{j=1}^{n}(y_{ij} - y_{Ij})p_i(x_{j,\hat{\mathcal{I}}};\alpha^{(t)})\{1 - p_i(x_{j,\hat{\mathcal{I}}};\alpha^{(t)})\}.$$

Step 3.3: update $\alpha^{(t+1)} \leftarrow \alpha^{(t)} - \{H(\alpha^{(t)})\}^{-1} S(\alpha^{(t)})$;

end

Let $\hat{\alpha} \triangleq \alpha^{(T)}$ denote the resulting estimator, and combine $\hat{\alpha}$ with (15) and (16) to determine the resulting predictive model $\hat{p}_i(x)$ for $i = 1, \cdots, I$.

Under the testing data $\mathcal{V}$;

Step 4: Prediction

For a new predictor $\tilde{x}$ in $\mathcal{V}$, use $\hat{p}_i(x)$ with $i = 1, \cdots, I$ to compute the corresponding probabilities $\tilde{\hat{p}}_1, \cdots, \tilde{\hat{p}}_I$. The predicted class i^* is then determined by $i^* = \underset{i\in\{1,\cdots,I\}}{\text{argmax}}\ \tilde{\hat{p}}_i$.

3.4 Logistic regression with heterogeneous network structured in predictors

We now present an alternative method to that described in Section 3.3. Instead of pooling all the predictors to feature the predictor network structure, this method, called the *logistic regression with heterogeneous network structured in predictors* (LR-HeteNet), stratifies the predictor information by class when characterizing the predictor network structures. The implementation is summarized in Algorithm 2.

Algorithm 2: LR-HeteNet

Under the training data $\mathcal{T}$;

for $i = 1, 2, \cdots, I$ **do**

Step 0: Let Y^i denote an n-dimensional vector formulated by (17).

Step 1: Class-dependent active set

Apply (18) to do feature screening and retain $\left[\frac{n_i}{\log n_i}\right]$ predictors among p-dimensional predictors. A set of selected predictors for class i is given by (19).

Step 2: Class-dependent predictor network

Based on selected predictors in $\hat{\mathcal{J}}_i$, use (8) to determine pairwise dependence structure and obtain (11). Denote $\hat{E}^i$ as the resulting network structure.

Step 3: Class-dependent predictive model

Given a initial value $\gamma_i^{(0)}$, then perform the Newton-Raphson algorithm;

for *step t with t* = 1, 2, ⋯, *T, say T* = 1000 **do**

Step 3.1: calculate the score function evaluated at the *t*th iterated value:

$$S_i(\gamma_{i,(t)}) \triangleq \frac{\partial \mathcal{L}_i(\gamma_i)}{\partial \gamma_i}\bigg|_{\gamma_i=\gamma_i^{(t)}} = \sum_{j=1}^{n_i} x_{j,\hat{\mathcal{J}}_i}\{y_j^i - \pi_i(x_{j,\hat{\mathcal{J}}_i};\gamma_i^{(t)})\},$$

where $\pi_i(x_{j,\hat{\mathcal{J}}_i};\gamma_i^{(t)})$ is (20) with parameters replaced by $\gamma_i^{(t)}$;

Step 3.2: calculate the Henssian matrix evaluated at the *t*th iterated value:

$$H_i(\gamma_i^{(t)}) \triangleq \frac{\partial^2 \mathcal{L}_i(\gamma_i)}{\partial \gamma_i \partial \gamma_i^\top}\bigg|_{\gamma_i=\gamma_i^{(t)}} = \sum_{j=1}^{n_i} x_{j,\hat{\mathcal{J}}_i} x_{j,\hat{\mathcal{J}}_i}^\top \pi_i(x_{j,\hat{\mathcal{J}}_i};\gamma_i^{(t)})\{1-\pi_i(x_{j,\hat{\mathcal{J}}_i};\gamma_i^{(t)})\}.$$

Step 3.3: update $\gamma_i^{(t+1)} \leftarrow \gamma_i^{(t)} - \{H_i(\gamma_i^{(t)})\}^{-1} S_i(\gamma_i^{(t)})$;

end

Let $\hat{\gamma} \triangleq \gamma_i^{(T)}$ denote the resulting estimator, and combine $\hat{\gamma}$ and (22) to determine the resulting predictive model $\hat{\pi}_i(x)$.

end

Under the testing data $\mathcal{V}$;

Step 4: Prediction

For a new predictor $\tilde{x}$ in $\mathcal{V}$, we use $\hat{\pi}_i(x)$ with i = 1, ⋯, I to compute the corresponding probabilities $\tilde{\hat{\pi}}_1, \cdots, \tilde{\hat{\pi}}_I$. The predicted class i^* is then determined by $i^* = \underset{i\in\{1,\cdots,I\}}{\operatorname{argmax}} \tilde{\hat{\pi}}_i$.

Be more specific, under the training data $\mathcal{T}$, we first introduce a binary, surrogate response variable for every $i = 1, \cdots, I$ and $j = 1, \cdots, n$. Let

$$Y_j^i = \begin{cases} 1, & Y_j = i \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

and let $Y^i = (0, \cdots, 0, Y_1^i, \cdots, Y_{n_i}^i, 0, \cdots, 0)^\top$ be an n-dimensional vector whose elements corresponding to class i are respectively $Y_1^i, \cdots, Y_{n_i}^i$, and the other elements are zero. That is, $Y^i = (\underbrace{0, \cdots, 0}_{n_1+\cdots+n_{i-1}}, \underbrace{1, \cdots, 1}_{n_i}, \underbrace{0, \cdots, 0}_{n_{i+1}+\cdots+n_I})^\top$ with $i = 1, \cdots, I$.

After that, we adopt the similar strategy in Algorithm 1 to construct predictive models for class i. Specifically, in Step 1 of Algorithm 2, let

$$\mathcal{J}_i = \{k : X_{\bullet k} \text{ is dependent on } Y^i\}$$

denote the true active set of the class i which contains all relevant predictors for the response Y^i with $|\mathcal{J}_i| < n_i$. Following (2), the signal of $X_{\bullet k}$ and Y^i is defined as $\omega_k^i \triangleq \xi(X_{\bullet k}, Y^i)$, and it can be estimated by

$$\hat{\omega}_k^i \triangleq \hat{\xi}(X_{\bullet k}, Y^i) = 1 - \frac{n\sum_{j=1}^{n-1} |r_{j+1}^i - r_j^i|}{2\sum_{j=1}^{n} \ell_j^i (n - \ell_j^i)}, \tag{18}$$

where, for $j = 1, \cdots, n$, $\ell_j^i \triangleq \#\{l : Y_{(l)}^i \geq Y_{(j)}^i\}$ and $r_j^i \triangleq \#\{l : Y_{(l)}^i \leq Y_{(j)}^i\}$ with $Y_{(j)}^i$ being the

rearranged response according to the sort of the kth predictors $X_{\bullet k}$. Therefore, $\mathcal{J}_i$ can be estimated as

$$\hat{\mathcal{J}}_i = \{k : \hat{\omega}_k^i \geq c_i n^{-\kappa_i} \text{ for } k = 1, \cdots, p\}, \tag{19}$$

where c_i and $\kappa_i \in (0, 1/2)$ are some prespecified threshold values. Let $X_{j,\hat{\mathcal{J}}_i} = \{X_{j,k} : k \in \hat{\mathcal{J}}_i\}$ denote the vector of all the active predictors that depends on Y^i for the jth subject. Moreover, since Y^i is defined as the response with binary outcomes, similar derivations in [35] show that (18) is valid to measure the dependence between categorical and continuous variables, and the point-biserial correlation coefficient is a special case of (18).

In Step 2 of Algorithm 2, let $V^i \triangleq \hat{\mathcal{J}}_i$ denote the vertex set containing predictors that are dependent on the class $i = 1, \cdots, I$. We apply the procedure described in Section 3.2 to determine the network structure of predictors in the class i. Let $\hat{E}^i = \{(s, t) : \hat{\theta}_{st}^i \neq 0\}$ denote an estimated set of edges for the class i, where $\hat{\theta}_{st}^i$ is the estimate of θ_{st} derived from (8) based on using the predictor measurements in the class i.

After that, Step 3 in algorithm 2 aims to fit a logistic regression model using the surrogate response vector Y^i with the estimated predictors network structure $\hat{E}^i$ incorporated for $i = 1, \cdots, I$. Specifically, for the jth component of Y^i, say Y_j^i, define $\pi_i(x_{j,\hat{\mathcal{J}}_i}) = P(Y_j^i = 1 | X_{j,\hat{\mathcal{J}}_i} = x_{j,\hat{\mathcal{J}}_i})$ and consider the parametric logistic regression model

$$\pi_i(x_{j,\hat{\mathcal{J}}_i}) \triangleq \pi_i(x_{j,\hat{\mathcal{J}}_i}; \gamma_i) = \frac{\exp\left(\gamma_{i0} + \sum_{(s,t)\in\hat{E}^i} \gamma_{i,st} x_{j,s} x_{j,t}\right)}{1 + \exp\left(\gamma_{i0} + \sum_{(s,t)\in\hat{E}^i} \gamma_{i,st} x_{j,s} x_{j,t}\right)}, \tag{20}$$

where $j = 1, \cdots, n$, $\gamma_i \triangleq (\gamma_{i0}, \gamma_{i\bullet}^\top)$ with $\gamma_{i\bullet} = (\gamma_{i,st} : (s, t) \in \hat{E}^i)^\top$ is the vector of parameters associated with class i. In the spirit of the maximum likelihood estimation (MLE) method (e.g., [42]), the log-likelihood function of (20) is given by

$$\mathcal{L}_i(\gamma_i) = \sum_{j=1}^{n_i} [y_j^i \pi_i(x_{j,\hat{\mathcal{J}}_i}; \gamma_i) + (1 - y_j^i)\{1 - \pi_i(x_{j,\hat{\mathcal{J}}_i}; \gamma_i)\}], \tag{21}$$

and the estimator of γ_i, denoted $\hat{\gamma}_i \triangleq (\hat{\gamma}_{i0}, \hat{\gamma}_{i\bullet}^\top)$, is obtained by maximizing (21). In applications, we implement the Newton-Raphson algorithm to obtain $\hat{\gamma}_i$; the detailed procedure is summarized in Algorithm 2. Consequently, for the realization $x_{j,\hat{\mathcal{J}}_i}$ of the $|\hat{\mathcal{J}}_i|$-dimensional vector $X_{j,\hat{\mathcal{J}}_i}$, based on (20), $\pi_i(x_{j,\hat{\mathcal{J}}_i})$ can be estimated by

$$\hat{\pi}_i(x_{j,\hat{\mathcal{J}}_i}) \triangleq \pi_i(x_{j,\hat{\mathcal{J}}_i}; \hat{\gamma}_i) = \frac{\exp\left(\hat{\gamma}_{i0} + \sum_{(s,t)\in\hat{E}^i} \hat{\gamma}_{i,st} x_{j,s} x_{j,t}\right)}{1 + \exp\left(\hat{\gamma}_{i0} + \sum_{(s,t)\in\hat{E}^i} \hat{\gamma}_{i,st} x_{j,s} x_{j,t}\right)} \tag{22}$$

for $i = 1, \cdots, I$.

Finally, when predictive models based on the training data $\mathcal{T}$ are obtained, we now examine the prediction for the testing data $\mathcal{V}$ in Step 4 of Algorithm 2. Let $\tilde{x}_{j,\hat{\mathcal{J}}_i}$ denote a $|\hat{\mathcal{J}}_i|$-dimensional predictor vector for a new subject. We calculate (22) with $x_{j,\hat{\mathcal{J}}_i}$ replaced by $\tilde{x}_{j,\hat{\mathcal{J}}_i}$ for $i = 1$,

$\cdots, I$, and let $\tilde{\tilde{\pi}}_1, \cdots, \tilde{\tilde{\pi}}_I$ denote the corresponding values. Let i^* denote the index which corresponds to the largest value of $\{\tilde{\tilde{\pi}}_1, \cdots, \tilde{\tilde{\pi}}_I\}$, i.e.,

$$\tilde{\tilde{\pi}}_{i^*} = \max_{i \in \{1,\cdots,I\}} \tilde{\tilde{\pi}}_i. \tag{23}$$

Then the class label for this new subject is predicted as i^*.

Remark 3.1 *The main difference between the MLR-HomoNet and LR-HeteNet methods is that the MLR-HomoNet method adopts the feature screening approach to retain informative predictors by pooling all subjects, while the feature screening approach of the LR-HeteNet method retains predictors under subjects that are in a specific class. It suggests that the estimated active sets* (19) *depend on the class and are different from each other, and thus, the resulting network structures determined by Step 2 of Algorithm 2 are different based on different classes. Therefore, we conclude that the MLR-HomoNet method only adopts different levels of gene expression values to classify tumor samples, while the LR-HeteNet method uses not only gene expression values but also class-dependent network structures to do the classification.*

4 Results

In this section, we aim to implement Algorithms 1 and 2 in Section 3 to the GCM dataset introduced in Section 2.1.

4.1 Detection of informative gene expressions via feature screening

In the GCM dataset, there are $I = 14$ classes. The dimension of predictors is $p = 16,063$ and the sample size is $n = 198$, where the size of the training set is 144 and the size of the testing set is 54. Following steps in Fig 1, we first implement the proposed method in Section 3 to fit models based on the training set, and then assess the performance of prediction by examining the testing set.

Since the dimension of predictors is extremely larger than the sample size, i.e., $p \gg n$, to determine the informative predictors, we adopt the screening signal (2) to retain informative gene expressions. The first strategy in Algorithm 1 is to apply (2) to evaluate the signal of $X_{\bullet k}$ and $Y \in \{1, \cdots, 14\}$ and determine the estimated active set (3); the second consideration in Algorithm 2 is to calculate the signal of $X_{\bullet k}$ and Y^i for $i = 1, \cdots, 14$ and then obtain the estimated class-dependent active set (19). As suggested in [33, 35, 36], under the training set, we consider to retain $\left[\frac{144}{\log(144)}\right] = 29$ gene expression values for the MLR-HomoNet method and retain $\left[\frac{n_i}{\log(n_i)}\right]$ gene expression values with $i = 1, \cdots, 14$ for the LR-HeteNet method, where n_i is the sample size of class i summarized in Table 1.

4.2 Network-based classification models

After the feature screening step, we next apply the estimation procedure in Section 3.2 to determine the network structure of selected gene expressions in the training set. Fig 2 displays the network structure with all samples accommodated, and the network structures of selected gene expressions based on different cancers are displayed in Fig 3. In Fig 2, we can see that the selected gene expressions have complex dependence structures. For example, gene expressions with ID `10111`, `9548`, and `9446` are connected with several gene expressions, while three gene expressions `10884`, `15854`, and `10208` have no connections with others. On the other hand, as shown in Fig 3, different classes have different selected gene expressions and associated network structures, which verifies the discussion in Remark 3.1. That is, as different kinds

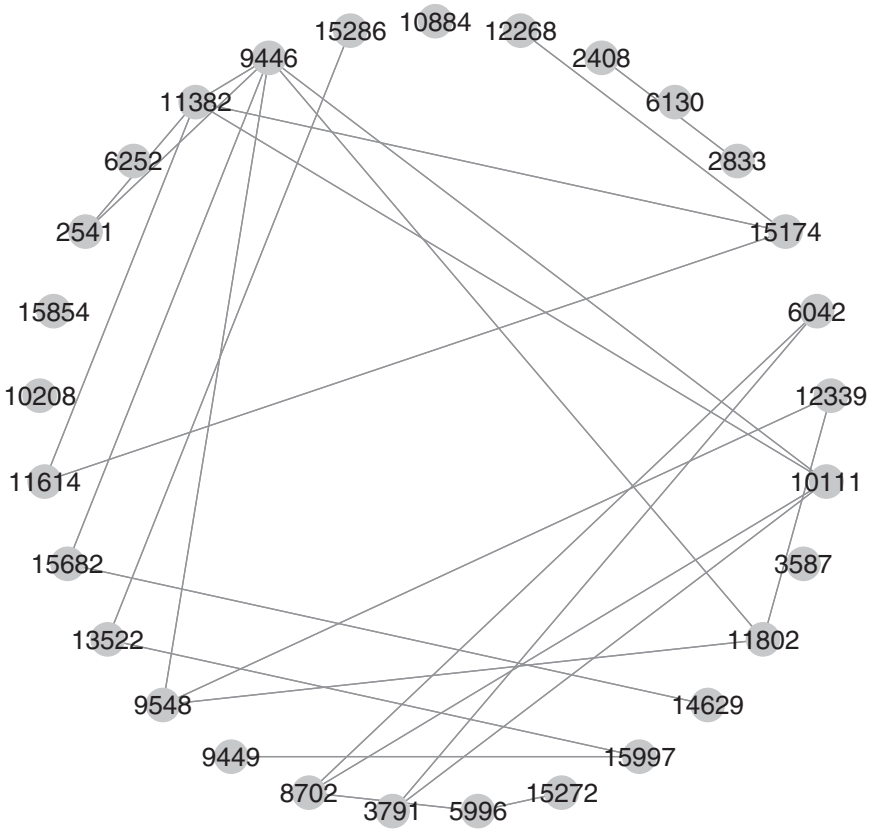

Fig 2. The whole network structure with selected gene expressions.

https://doi.org/10.1371/journal.pone.0274440.g002

of cancer differ in their corresponding gene expressions, according to the specific network structures of gene expressions produced from our analysis, we can infer which cancer each tumor sample is from.

To adopt the determined network structures to examine the classification, we implement the network structures and the training set to the classification models proposed in Sections 3.3 and 3.4, respectively. To see the fitness of two models, we first implement the training data to the fitted models and examine the classification. The 14×14 confusion matrices based on the MLR-HomoNet and LR-HeteNet methods are shown in Tables 2 and 3, respectively, where columns are labels from the training data $\mathcal{T}$, rows are labels of fitted values, diagonal entries reflect number of correct classification, and nondiagonal entries are number of misclassification by fitted values. In general, both methods show satisfactory model fittness as the accuracy of classification is high. Moreover, we observe that the LR-HeteNet method seems to slightly outperform the MLR-HomoNet method since the latter method produces slightly larger misclassification on BR, PR, CO, and UT than those of the former method. This result makes sense because the LR-HeteNet method is based on class-dependent network structure that can directly reflect the corresponding cancers. For a clear visualization, we further display two heatmaps in Fig 4, which are obtained by Tables 2 and 3 with each row divided by the class-dependent sample size in the training data. We observe that diagonal entries have dark color, which indicate that the proportion of true classification is high and Algorithms 1 and 2 give well-fitted models.

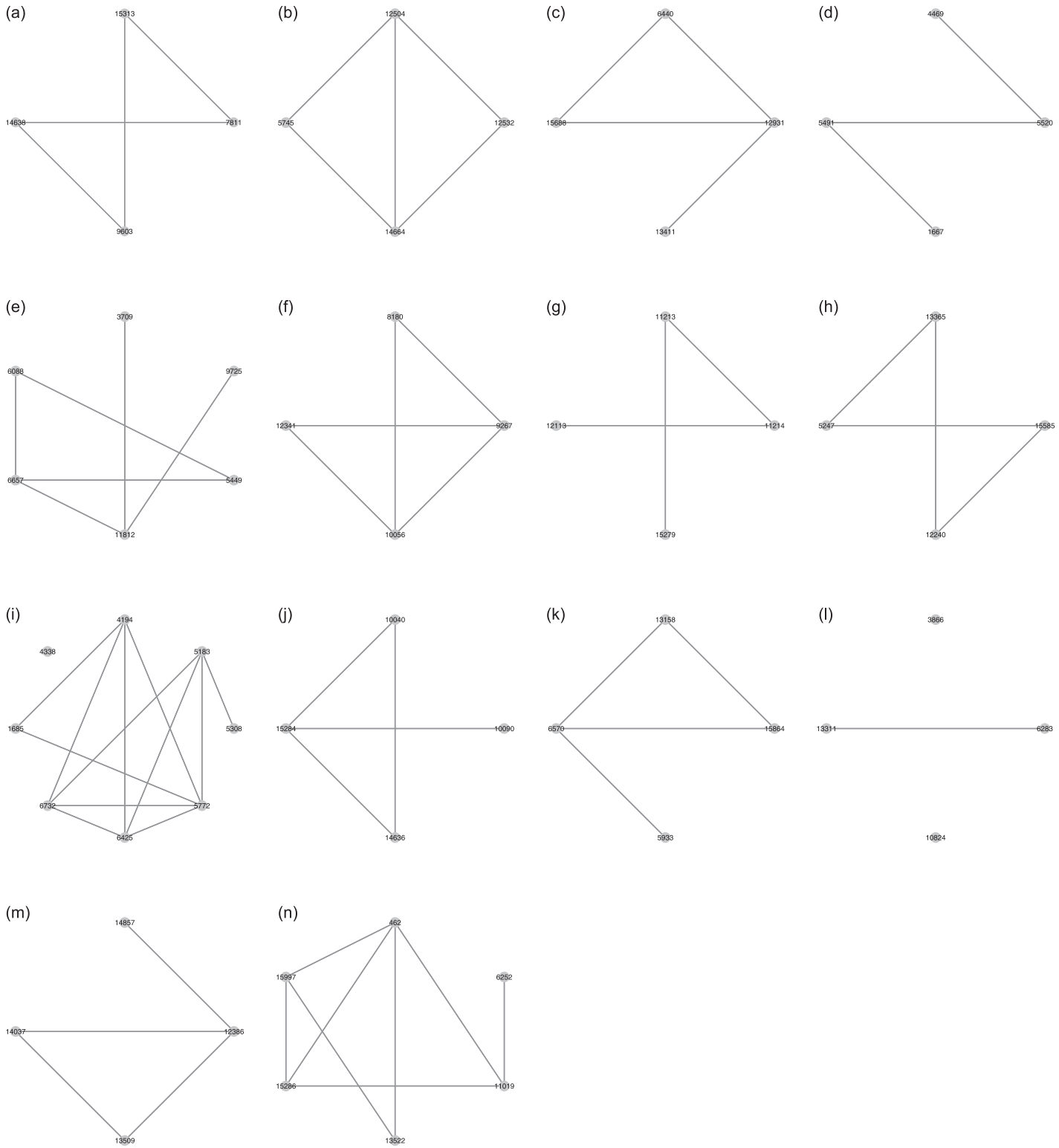

Fig 3. The network structure with selected gene expressions based on different cancers.

https://doi.org/10.1371/journal.pone.0274440.g003

Table 2. A 14 × 14 confusion matrix: Model fittness based on the MLR-HomoNet method for the training data $\mathcal{T}$.

	BR	PR	LE	CO	LU	BL	CNS	UT	LY	RE	PA	OV	ME	ML
BR	6	1	0	0	0	1	1	0	0	1	1	0	0	0
PR	0	6	0	0	0	0	0	1	0	0	0	0	0	0
LE	0	0	6	0	0	0	0	0	0	0	0	0	0	0
CO	1	0	0	7	0	1	0	1	0	0	0	0	0	0
LU	0	0	0	0	16	0	1	0	0	1	0	0	0	0
BL	0	0	0	0	0	6	0	0	0	0	0	1	0	0
CNS	0	1	0	1	0	0	6	0	0	0	0	0	0	0
UT	1	0	1	0	0	0	0	5	0	0	0	0	0	0
LY	0	0	0	0	0	0	0	0	24	0	0	0	0	0
RE	0	0	0	0	0	0	0	0	0	6	0	0	0	0
PA	0	0	0	0	0	0	0	1	0	0	7	0	1	0
OV	0	0	1	0	0	0	0	0	0	0	0	7	0	0
ME	0	0	0	0	0	0	0	0	0	0	0	0	7	0
ML	0	0	0	0	0	0	0	0	0	0	0	0	0	16

https://doi.org/10.1371/journal.pone.0274440.t002

4.3 Prediction

When the predictive models are constructed, we now assess the performance of the proposed method by examining the prediction for the testing data. We implement the predictors in the testing data to the two proposed methods, and then make the prediction of classification. After that, we summarize the response in the testing data and the predictive classes to 14 × 14 confusion matrices in Tables 4 and 5, respectively, where columns are labels from the testing samples $\mathcal{V}$, rows are labels of predicted values, diagonal entries reflect number of correct classification, and nondiagonal entries are number of misclassification by predicted values. Moreover, we also display two heatmaps in Fig 5 that are obtained by Tables 4 and 5 with each row divided by the class-dependent sample size in the testing data. From confusion matrices and heatmaps, We can see that two proposed methods have satisfactory performance in prediction because most of predicted classes are the same as class labels in the testing data, except for little misclassification.

Table 3. A 14 × 14 confusion matrix: Model fittness based on the LR-HeteNet method for the training data $\mathcal{T}$.

	BR	PR	LE	CO	LU	BL	CNS	UT	LY	RE	PA	OV	ME	ML
BR	7	0	0	0	0	0	1	0	0	0	0	0	1	0
PR	0	7	0	0	0	0	0	0	0	1	0	0	0	0
LE	1	0	6	0	0	0	0	0	0	0	1	0	0	0
CO	0	0	0	8	0	1	0	0	0	0	0	0	0	0
LU	0	0	0	0	16	0	1	0	0	1	0	0	0	0
BL	0	0	0	0	0	7	0	0	0	0	0	0	0	0
CNS	0	1	0	0	0	0	6	0	0	0	0	0	0	0
UT	0	0	1	0	0	0	0	7	0	0	0	0	0	0
LY	0	0	0	0	0	0	0	0	24	0	0	0	0	0
RE	0	0	0	0	0	0	0	0	0	6	0	0	0	0
PA	0	0	0	0	0	0	0	1	0	0	7	1	0	0
OV	0	0	1	0	0	0	0	0	0	0	0	7	0	0
ME	0	0	0	0	0	0	0	0	0	0	0	0	7	0
ML	0	0	0	0	0	0	0	0	0	0	0	0	0	16

https://doi.org/10.1371/journal.pone.0274440.t003

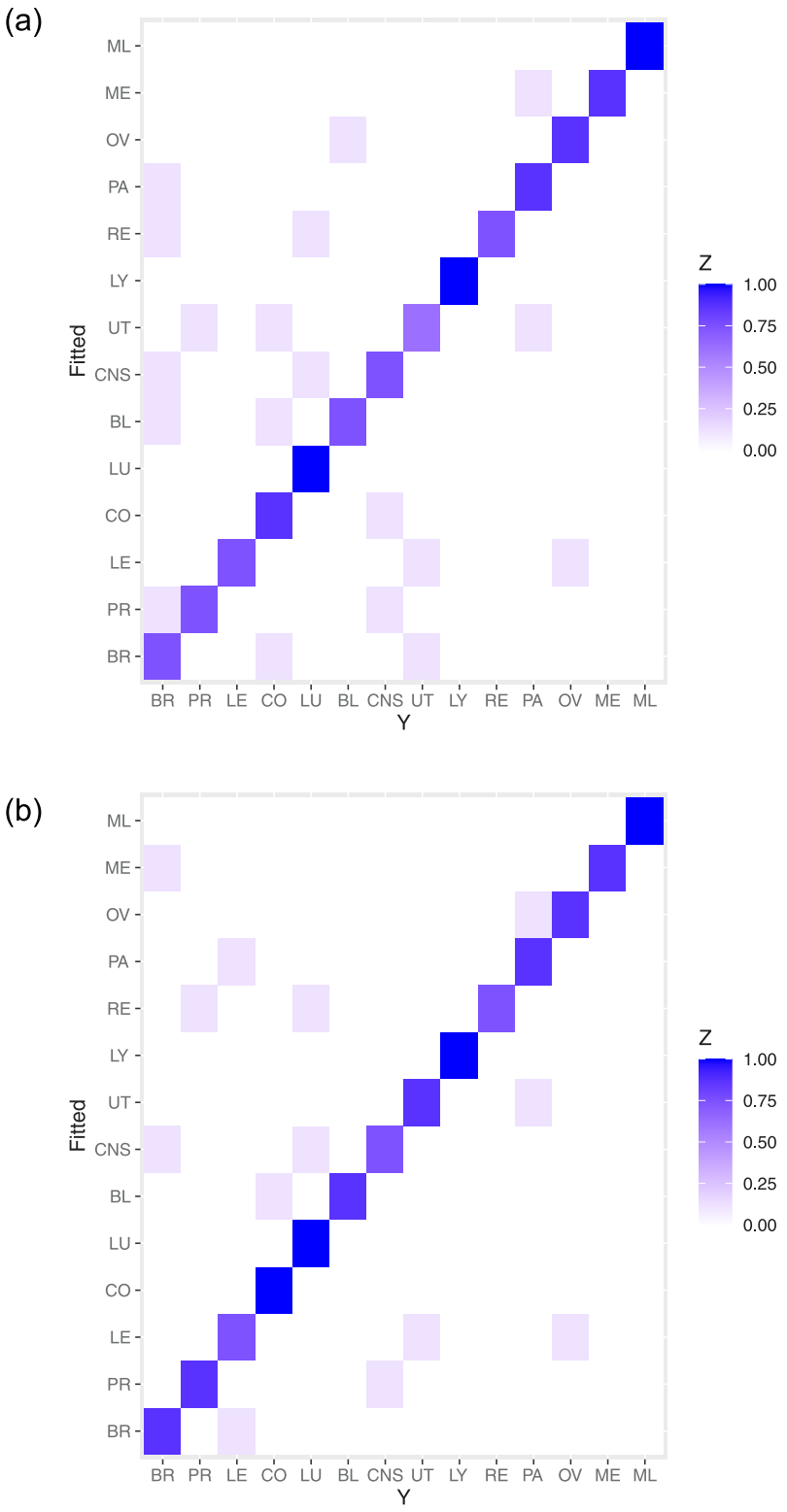

Fig 4. Heatmaps for the fitted values based on two proposed methods under the training data. The left panel is obtained by Algorithm 1, the right panel is obtained by Algorithm 2. *Z* represents the proportion of (mis)classification.

https://doi.org/10.1371/journal.pone.0274440.g004

Table 4. A 14 × 14 confusion matrix: Prediction based on the MLR-HomoNet method for the testing data $\mathcal{V}$.

	BR	PR	LE	CO	LU	BL	CNS	UT	LY	RE	PA	OV	ME	ML
BR	2	0	0	0	0	0	0	0	0	1	0	0	0	0
PR	0	4	0	0	1	0	0	0	0	0	0	0	0	0
LE	1	0	2	0	0	0	0	0	0	0	0	0	0	0
CO	0	0	0	3	0	0	0	0	1	0	0	0	0	0
LU	0	0	0	1	3	0	0	0	0	0	0	0	0	0
BL	0	0	0	0	0	3	0	0	0	0	0	0	0	0
CNS	0	0	0	0	0	0	2	0	0	0	0	0	0	0
UT	0	0	0	0	0	0	0	3	0	0	0	1	0	0
LY	0	0	0	0	0	0	0	0	5	0	0	0	0	0
RE	0	0	0	0	0	0	0	0	0	3	0	0	0	0
PA	0	0	0	0	0	0	0	0	0	0	3	0	0	0
OV	0	0	0	0	0	1	0	0	0	0	0	5	0	1
ME	0	0	0	0	0	0	0	0	0	0	0	0	3	0
ML	0	0	0	0	0	0	0	0	0	0	0	0	0	5

https://doi.org/10.1371/journal.pone.0274440.t004

To assess the performance of classification and prediction numerically, we evaluate some commonly used criteria: micro averaged metrics, macro averaged metrics, and the adjusted Rand index. For a subject j in the testing data with $j = 1, \cdots, 54$, let $\hat{y}_{\text{new},j}$ denote the predicted class label determined by the prediction models and let $y_{\text{new},j}$ denote the class label in the testing data. For class $i = 1, \cdots, I$, we respectively calculate the number of the true positives (TP), the number of the false positives (FP), and the number of the false negatives (FN) as

$$\text{TP}_i = \sum_{j=1}^{54} \mathbb{I}(y_{\text{new},j} = i, \hat{y}_{\text{new},j} = i), \tag{24}$$

$$\text{FP}_i = \sum_{j=1}^{54} \mathbb{I}(y_{\text{new},j} \neq i, \hat{y}_{\text{new},j} = i), \tag{25}$$

Table 5. A 14 × 14 confusion matrix: Prediction based on the LR-HeteNet method for the testing data $\mathcal{V}$.

	BR	PR	LE	CO	LU	BL	CNS	UT	LY	RE	PA	OV	ME	ML
BR	2	0	0	0	0	0	0	0	1	0	0	0	0	0
PR	0	4	0	0	1	0	0	0	0	0	0	0	0	0
LE	1	0	2	0	0	0	0	0	0	0	0	0	0	0
CO	0	0	0	3	0	0	0	0	1	0	0	0	0	0
LU	0	0	0	1	3	0	0	0	0	0	0	0	0	0
BL	0	0	0	0	0	4	0	0	0	0	0	0	0	0
CNS	0	0	0	0	0	0	2	0	0	0	0	0	0	0
UT	0	0	0	0	0	0	0	3	0	0	0	0	0	0
LY	0	0	0	0	0	0	0	0	4	0	0	0	0	0
RE	0	0	0	0	0	0	0	0	0	4	0	0	0	0
PA	0	0	0	0	0	0	0	0	0	0	3	0	0	0
OV	0	0	0	0	0	0	0	0	0	0	0	6	0	0
ME	0	0	0	0	0	0	0	0	0	0	0	0	3	0
ML	0	0	0	0	0	0	0	0	0	0	0	0	0	6

https://doi.org/10.1371/journal.pone.0274440.t005

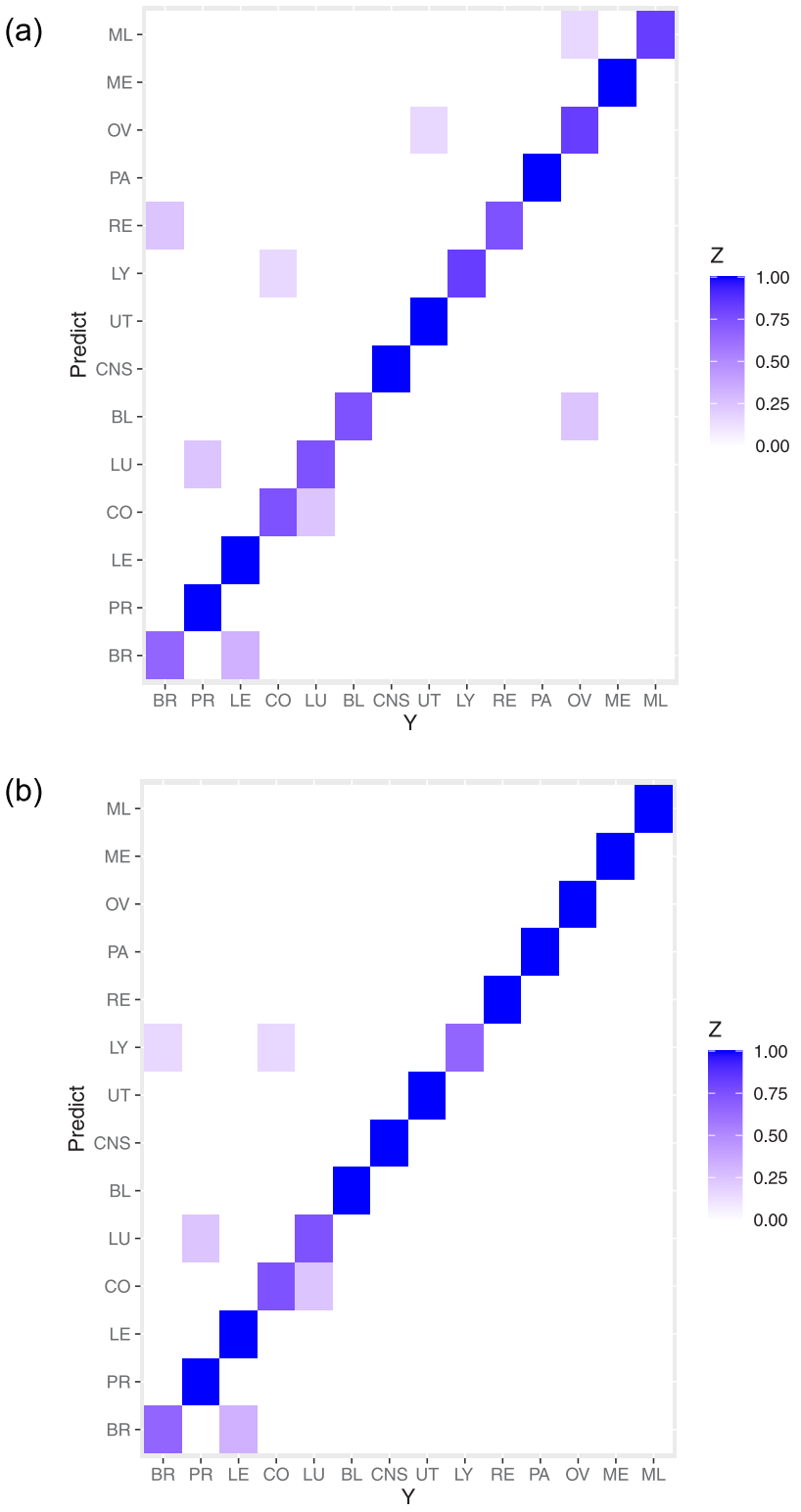

Fig 5. Heatmaps for the predicted values based on two proposed methods under the testing data. The left panel is obtained by Algorithm 1, the right panel is obtained by Algorithm 2. *Z* represents the proportion of (mis)classification.

https://doi.org/10.1371/journal.pone.0274440.g005

and

$$\mathrm{FN}_i = \sum_{j=1}^{54} \mathbb{I}(y_{\mathrm{new},j} = i, \hat{y}_{\mathrm{new},j} \neq i). \tag{26}$$

For micro averaged metrics, precision and recall are, respectively, defined in terms of (24), (25), and (26):

$$\mathrm{PRE}_{\mathrm{micro}} = \frac{\sum_{i=1}^{I} \mathrm{TP}_i}{\sum_{i=1}^{I} \mathrm{TP}_i + \sum_{i=1}^{I} \mathrm{FP}_i} \tag{27}$$

and

$$\mathrm{REC}_{\mathrm{micro}} = \frac{\sum_{i=1}^{I} \mathrm{TP}_i}{\sum_{i=1}^{I} \mathrm{TP}_i + \sum_{i=1}^{I} \mathrm{FN}_i}. \tag{28}$$

Then Micro-F-score is defined as

$$\mathrm{F}_{\mathrm{micro}} = 2 \times \frac{\mathrm{PRE}_{\mathrm{micro}} \times \mathrm{REC}_{\mathrm{micro}}}{\mathrm{PRE}_{\mathrm{micro}} + \mathrm{REC}_{\mathrm{micro}}}. \tag{29}$$

On the other hand, for macro averaged metrics, for $i = 1, \cdots, I$, let $\mathrm{PRE}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}$ denote precision for class i, and let $\mathrm{REC}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}$ denote recall for class i. Then the overall precision and recall are, respectively, given by

$$\mathrm{PRE}_{\mathrm{macro}} = \frac{1}{I} \sum_{i=1}^{I} \mathrm{PRE}_i \tag{30}$$

and

$$\mathrm{REC}_{\mathrm{macro}} = \frac{1}{I} \sum_{i=1}^{I} \mathrm{REC}_i; \tag{31}$$

and Macro-F-score is defined as

$$\mathrm{F}_{\mathrm{macro}} = 2 \times \frac{\mathrm{PRE}_{\mathrm{macro}} \times \mathrm{REC}_{\mathrm{macro}}}{\mathrm{PRE}_{\mathrm{macro}} + \mathrm{REC}_{\mathrm{macro}}}. \tag{32}$$

According to the definitions, when all subjects are correctly classified, then FP and FN are equal to zero, yielding that PRE and REC are equal to one; if all subjects are falsely classified, then TP is equal to zero, and thus, PRE and REC are equal to zero. Therefore, values of PRE and REC are between zero to one. Moreover, the F-score falls in [0, 1] as well by treating 0/0 as zero. In principle, higher values of PRE, REC and F-score based on both micro and macro reflect better performance of methods ([20–22]).

In addition to criteria above, the other commonly used criterion is the adjusted Rand index (ARI). For $i, l = 1, \cdots, I$, let $n_{il} = \sum_{j=1}^{n} \mathbb{I}(y_{\mathrm{new},j} = i, \hat{y}_{\mathrm{new},j} = l)$. Moreover, define $a_i = \sum_{l=1}^{I} n_{il}$ for

Table 6. A list of existing methods and corresponding packages.

Method[Reference]	Function	R Package
SVM [44]	`svm`	`e1071`
KNN [45]	`kNN`	`DMwR`
LDA [46]	`lda`	`MASS`
Bayes [44]	`naiveBayes`	`e1071`
ANN [47]	`neuralnet`	`neuralnet`
XGBoost [48]	`xgb.train`	`xgboost`
RF [49]	`randomForest`	`randomForest`
Bagging [50]	`ipredbagg`	`ipred`
LSTM [51]	`trainr`	`rnn`

https://doi.org/10.1371/journal.pone.0274440.t006

$i = 1, \cdots, I$ and $b_l = \sum_{i=1}^{I} n_{il}$ for $l = 1, \cdots, I$. Then ARI is defined as (e.g., [43])

$$\mathrm{ARI} = \frac{\sum_{i,l=1}\binom{n_{il}}{2} - \left\{\sum_i \binom{a_i}{2}\sum_l \binom{b_l}{2}\right\}\Big/\binom{n}{2}}{\left\{\sum_i \binom{a_i}{2} + \sum_l \binom{b_l}{2}\right\}\Big/2 - \left\{\sum_i \binom{a_i}{2}\sum_l \binom{b_l}{2}\right\}\Big/\binom{n}{2}}. \tag{33}$$

As mentioned in [43], ARI is bounded above by one, and higher value of ARI indicates accurate classification.

We primarily adopt (27), (28), (29), (30), (31), (32), and (33) to assess the performance of two proposed methods. In addition, to compare with the proposed methods, we also examine several well established supervised learning methods, including logistic regression models *without* incorporating network structure [42], the support vector machine (SVM) that was examined by [30], K-nearest neighbor (KNN), linear discriminant analysis (LDA), Bayes, artificial neural network (ANN), XGBoost, random forest (RF), bagging, and long short-term memory (LSTM) methods. The implementation of corresponding R packages is summarized in Table 6.

The prediction results of the proposed and competitive methods are summarized in Table 7. In general, we can observe that the two proposed methods have the largest values of

Table 7. Prediction of classification for the testing data $\mathcal{V}$.

Method	$\mathrm{PRE}_{\mathrm{micro}}$	$\mathrm{REC}_{\mathrm{micro}}$	$\mathrm{F}_{\mathrm{micro}}$	$\mathrm{PRE}_{\mathrm{macro}}$	$\mathrm{REC}_{\mathrm{macro}}$	$\mathrm{F}_{\mathrm{macro}}$	ARI
Agresti	0.693	0.697	0.695	0.688	0.696	0.692	0.453
SVM	0.801	0.812	0.806	0.813	0.820	0.816	0.786
LDA	0.705	0.705	0.705	0.699	0.694	0.696	0.474
KNN	0.677	0.663	0.670	0.654	0.666	0.660	0.433
Bayes	0.837	0.838	0.838	0.840	0.838	0.839	0.804
ANN	0.844	0.845	0.844	0.844	0.844	0.844	0.821
XGBoost	0.816	0.818	0.817	0.820	0.816	0.818	0.797
RF	0.840	0.838	0.839	0.842	0.841	0.841	0.813
Bagging	0.840	0.836	0.838	0.841	0.841	0.841	0.809
LSTM	0.835	0.837	0.836	0.837	0.840	0.838	0.794
MLR-HomoNet	0.856	0.871	0.863	0.867	0.878	0.872	0.833
LR-HeteNet	0.884	0.896	0.890	0.903	0.910	0.906	0.856

https://doi.org/10.1371/journal.pone.0274440.t007

PRE, REC, F-score, and ARI than other existing methods. For the comparisons among existing methods, we can see that advanced machine learning or deep learning methods (e.g., ANN, RF, Bagging) outperform the conventional ones, such as LDA or SVM, but are less satisfactory than the proposed methods because of slightly large misclassification. It verifies that incorporating network structures would improve the accuracy of classification and prediction. In addition, the other reason is that, unlike existing methods that possibly incur overfitting because of direct implementation of all gene expression values to fit models, the two proposed methods simply retain gene expression values and detect network structures that are related to the response, yielding parsimonious models. In this way, noises and impacts induced by irrelevant gene expression values can be eliminated. Compared with two proposed methods, we can see that the LR-HeteNet method outperforms the MLR-HomoNet method with larger values of criteria. The main reason is that the MLR-HomoNet model in Section 3.3 directly deals with multi-label classification by using a common network structure to classify tumors to the corresponding cancers. To simultaneously reflect information to all classes, the network structure displayed in Fig 2 is expected to require more gene expression values and complex interactions. On the other hand, the LR-HeteNet method in Section 3.4 identifies predictors and unique network structure to reflect a specific cancer, suggesting that types of cancers can be uniquely represented by different network structures of gene expression values. As shown in Fig 3, one can directly adopt a given network structure to classify tumors to their cancers with high accuracy of prediction. In summary, with noise induced by irrelevant predictors removed and informative network structures of predictors accommodated, the accuracy of classification and prediction has significant improvement.

5 Discussion

In this paper, we present the network-based classification method to predict the classification of the tumor samples, which is an ultrahigh dimensional system, i.e., with multitudinous gene expressions as predictors. In the proposed method, we first adopt model-free feature screening technique to retain informative gene expressions from ultrahigh-dimensional data. After that, we identify the network structures of the detected gene expressions based on different cancers, and the property of the network structure recovery allows us to fit the nominal logistic regression based on the network structure and examine the classification and prediction. Compared with other existing methods, the proposed method gives more precise prediction results.

There are several possible extensions based on the current work. For example, the RNA sequences, regarded as count data, are also frequently explored in bioinformatics. The proposed method can be naturally extended to deal with the RNA sequence data by treating them as the predictors because the signal of detecting predictors (2) is free of distribution of random variables, and the identification of network structure in Section 3.2 is based on exponential family graphical models. For the implementation of classification models, it is interesting to explore other machine learning methods, such as SVM, LDA, or KNN, and other deep learning approaches that are popular in data science.

Moreover, the research gap still exists and more explorations can be done by extending the proposed method. For example, as discussed in [32], measurement error in predictors is ubiquitous in data analysis, especially that mismeasurement is inevitable in gene expression data (e.g, [52]). Ignoring measurement error effects is expected to increase the possibility of false classification and lead to wrong conclusion. Therefore, it is important to develop a new error-eliminating strategy to deal with measurement error based on the current method. Finally, as R packages associated with some of the existing methods have been developed, the new method proposed here anticipates a corresponding R package.

Supporting information

S1 Data.
(ZIP)

Acknowledgments

The author would like to appreciate Lingyu Cai for technical support of programming code, helpful language editing, grammar revision, and proofreading. The author thanks the editorial team for providing constructive and suggestive comments to improve the presentation of the manuscript.

Author Contributions

Conceptualization: Li-Pang Chen.

Formal analysis: Li-Pang Chen.

Investigation: Li-Pang Chen.

Methodology: Li-Pang Chen.

Resources: Li-Pang Chen.

Visualization: Li-Pang Chen.

Writing – original draft: Li-Pang Chen.

References

1. Gálvez J. M., Castillo D., Herrera L. J., San Román B., Valenzuela O., Ortuno F. M., et al. (2018). Multi-class classification for skin cancer profiling based on the integration of heterogeneous gene expression series. *PLoS ONE*, 13(5), e0196836. https://doi.org/10.1371/journal.pone.0196836 PMID: 29750795
2. Lee Y. and Lee C.-K. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*. 19, 1132–1139. https://doi.org/10.1093/bioinformatics/btg102 PMID: 12801874
3. Cristianini N. and Shawe-Taylor J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge.
4. Huang M. W., Chen C. W., Lin W. C., Ke S. W., and Tsai C. F. (2017). SVM and SVM ensembles in breast cancer prediction. *PLoS ONE*, 12(1), e0161501. https://doi.org/10.1371/journal.pone.0161501 PMID: 28060807
5. Guo Y., Hastie T., Tibshirani R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8, 86–100. https://doi.org/10.1093/biostatistics/kxj035 PMID: 16603682
6. Safo S. E. and Ahn J. (2016). General sparse multi-class linear discriminant analysis. *Computational Statistics and Data Analysis*, 99, 81–90. https://doi.org/10.1016/j.csda.2016.01.011
7. Hastie T., Tibshirani R., and Friedman J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York.
8. James G., Witten D., Hastie T., and Tibshirani R. (2017). *An Introduction to Statistical Learning: with Applications in R*. Springer, New York.
9. Chen L.-P. (2019). Foundations of Machine Learning by Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Statistical Papers*, 60, 1793–1795. https://doi.org/10.1007/s00362-019-01124-9
10. Heenaye-Mamode Khan M., Boodoo-Jahangeer N., Dullull W., Nathire S., Gao X., Sinha G. R., et al. (2021). Multi-class classification of breast cancer abnormalities using Deep Convolutional Neural Network (CNN). *PLoS ONE*, 16(8), e0256500. https://doi.org/10.1371/journal.pone.0256500 PMID: 34437623
11. Pandey A. and Roy S. S. (2022). Protein sequence classification using convolutional neural network and natural language processing. *Handbook of Machine Learning Applications for Genomics, edited by S. S. Roy and Y. H. Taguchi*, 133–144.

12. Roy S. S., Samui P., Deo R., and Ntalampiras S. (Eds.). (2018). *Big Data in Engineering Applications* (Vol. 44). Springer, Berlin/Heidelberg, Germany.

13. Roy S. S. and Taguchi Y. H. (2022). *Handbook of Machine Learning Applications for Genomics*. Springer Nature, Singapore.

14. Samui P., Roy S. S., and Balas V. E. (Eds.). (2017). *Handbook of Neural Computation*, Academic Press.

15. Zhu S. X. Y. and Pan W. (2009). Network-based support vector machine for classification of microarray samples. *BMC Bioinformatics*, 10, 1–11. https://doi.org/10.1186/1471-2105-10-S1-S21 PMID: 19208121

16. Zi X., Liu Y., Gao P. (2016). Mutual information network-based support vector machine for identification of rheumatoid arthritis-related genes. *International Journal of Clinical and Experimental Medicine*, 9, 11764–11771.

17. Cai W., Guan G., Pan R., Zhu X., and Wang H. (2018). Network linear discriminant analysis. *Computational Statistics and Data Analysis*, 117, 32–44. https://doi.org/10.1016/j.csda.2017.07.007

18. Huttenhower C., Flamholz A.I., Landis J.N. et al. (2007). Nearest neighbor networks: clustering expression data based on gene neighborhoods. *BMC Bioinformatics*, 8, 250, 1–13. https://doi.org/10.1186/1471-2105-8-250 PMID: 17626636

19. He, W., Yi, G. Y., and Chen, L.-P. (2019). Support vector machine with graphical network structures in features. *Proceedings*, *Machine Learning and Data Mining in Pattern Recognition*, *15th International Conference on Machine Learning and Data Mining*, *MLDM 2019*, *vol.II*, New York, NY, USA, ibai-publishing, 557–570.

20. Chen L.-P., Yi G. Y., Zhang Q., and He W. (2019). Multiclass analysis and prediction with network structured covariates. *Journal of Statistical Distributions and Applications*, 6:6. https://doi.org/10.1186/s40488-019-0094-2

21. Chen L.-P. (2022a). Network-based discriminant analysis for multiclassification. *Journal of Classification*. To appear. https://doi.org/10.1007/s00357-022-09414-y

22. Chen L.-P. (2022b). Nonparametric discriminant analysis with network structures in predictor. *Journal of Statistical Computation and Simulation*. To appear. https://doi.org/10.1080/00949655.2022.2084618

23. Baladanddayuthapani V., Talluri R., Ji Y., Coombes K. R., Lu Y., Hennessy B. T., et al. (2014). Bayesian sparse graphical models for classification with application to protein expression data. *The Annals of Applied. Statistics*, 8, 1443–1468. https://doi.org/10.1214/14-AOAS722

24. Peterson C. B., Stingo F. C., and Vannucci M. (2015). Joint Bayesian variable and graph selection for regression models with network-structured predictors. *Statistics in Medicine*, 35, 1017–1031. https://doi.org/10.1002/sim.6792 PMID: 26514925

25. Roy S. S. and Taguchi Y. H. (2021). Identification of genes associated with altered gene expression and m6A profiles during hypoxia using tensor decomposition based unsupervised feature extraction. *Scientific Reports*, 11(1), 1–18. https://doi.org/10.1038/s41598-021-87779-7 PMID: 33903618

26. Tschodu D., Ulm B., Bendrat K., Lippoldt J., Gottheil P., Käs J. A., et al. (2022). Comparative analysis of molecular signatures reveals a hybrid approach in breast cancer: combining the Nottingham Prognostic Index with gene expressions into a hybrid signature. *PloS ONE*, 17(2), e0261035. https://doi.org/10.1371/journal.pone.0261035 PMID: 35143511

27. Zhang X., Wu Y., Wang L., and Li R. (2016). Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society*, *Series B*, 78, 53–76. https://doi.org/10.1111/rssb.12100 PMID: 26778916

28. Maugis C., Celeux G., and Martin-Magniette M.-L. (2011). Variable selection in model-based discriminant analysis. *Journal of Multivariate Analysis*, 102, 1374–1387. https://doi.org/10.1016/j.jmva.2011.05.004

29. Wang C., Cao L., and Miao B. (2013). Optimal feature selection for sparse linear discriminant analysis and its applications in gene expression data. *Computational Statistics and Data Analysis*, 66, 140–149. https://doi.org/10.1016/j.csda.2013.04.003

30. Ramaswamy S., Tamayo P., Rifkin R. et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States*, 98, 15149–15154. https://doi.org/10.1073/pnas.211566398 PMID: 11742071

31. Lukashin A. V., Lukashev M. E., and Fuchs R. (2003). Topology of gene expression networks as revealed by data mining and modeling. *Bioinformatics*, 19, 1909–1916. https://doi.org/10.1093/bioinformatics/btg333 PMID: 14555623

32. Chen L.-P. (2018). Multiclassification to gene expression data with some complex features. *Biostatistics and Biometrics Open Access Journal*, 9, 555751. https://doi.org/10.19080/BBOAJ.2018.09.555751

33. Fan J. and Lv J. (2008). Sure independence screening for ultra high dimensional feature space. *Journal of the Royal Statistical Society*, *Series B*, 70, 849–911. https://doi.org/10.1111/j.1467-9868.2008.00674.x

34. Chatterjee S. (2021). A new coefficient of correlation. *Journal of the American Statistical Association*, 16, 2009–2022. https://doi.org/10.1080/01621459.2020.1758115

35. Chen, L.-P. (2020). A note of feature screening via rank-based coefficient of correlation. arXiv:2008.04456

36. Chen L.-P. (2021). Feature screening based on distance correlation for ultrahigh-dimensional censored data with covariates measurement error. *Computational Statistics*, 36, 857–884. https://doi.org/10.1007/s00180-020-01039-2

37. Yang E., Ravikumar P., Allen G. I., and Liu Z. (2015). Graphical models via univariate exponential family distribution. *Journal of Machine Learning Research*, 16, 3813–3847. PMID: 27570498

38. Meinshausen N. and Bühlmann P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34, 1436–1462. https://doi.org/10.1214/009053606000000281

39. Schwarz G. (1978). Estimating the dimension of model. *Annals of Statistics*, 6, 461–464. https://doi.org/10.1214/aos/1176344136

40. Ravikumar P., Wainwright M. J., and Lafferty J. (2010). High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38, 1287–1319. https://doi.org/10.1214/09-AOS691

41. Agresti A. (2007). *An Introduction to Categorical Data Analysis*. Wiley, New York.

42. Agresti A. (2012). *Categorical Data Analysis*. Wiley, New York.

43. Hubert L. and Arabie P. (1985). Comparing partitions. *Journal of Classification*, 2, 193–218. https://doi.org/10.1007/BF01908075

44. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. et al. (2022). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-11. https://CRAN.R-project.org/package=e1071

45. Torgo, L. (2022). DMwR: Functions and data for "Data Mining with R". R package version 0.4.1. https://CRAN.R-project.org/package=DMwR

46. Ripley, B., Venables, B., Bates, D. M., Hornik, K. et al. (2022). MASS: Support functions and datasets for venables and Ripley's MASS. R package version 7.3-57. https://CRAN.R-project.org/package=MASS

47. Fritsch, S., Guenther, F., Wright, M. N., Suling, M., and Mueller, S. M. (2019). neuralnet: Training of neural networks. R package version 1.44.2. https://CRAN.R-project.org/package=neuralnet

48. Chen, T., He, T., Benesty, M., Khotilovich, V. et al. (2022). xgboost: Extreme gradient boosting. R package version 1.6.0.1. https://CRAN.R-project.org/package=xgboost

49. Breiman, L., Cutler, A., Liaw, A. and Wiener, M. (2022). randomForest: Breiman and Cutler's random forests for classification and regression. R package version 4.7-1. https://CRAN.R-project.org/package=randomForest

50. Peters, A., Hothorn, T., Ripley, B. D., Therneau, T., and Atkinson, B. (2022). ipred: Improved predictors. R package version 0.9-13. https://CRAN.R-project.org/package=ipred

51. Quast, B. and Fichou, D. (2022). rnn: Recurrent Neural Network. R package version 1.5.0. https://CRAN.R-project.org/package=rnn

52. Chen L.-P. and Yi G. Y. (2021). Analysis of Noisy Survival Data with Graphical Proportional Hazards Measurement Error Models. *Biometrics*, 77, 956–969. https://doi.org/10.1111/biom.13331 PMID: 32687216