Received: 16 April 2021 | Revised: 14 September 2021 | Accepted: 29 October 2021

DOI: 10.1002/sim.9267

RESEARCH ARTICLE

Statistics in Medicine WILEY

Feature selection and classification over the network with missing node observations

Zhuxuan Jin[1] | **Jian Kang**[2] | **Tianwei Yu**[3]

[1]Splunk Inc., San Francisco, California

[2]Department of Biostatistics, University of Michigan, Ann Arbor, Michigan

[3]School of Data Science and Warshel Institute, The Chinese University of Hong Kong - Shenzhen, and Shenzhen Research Institute of Big Data, Shenzhen, China

Correspondence
Jian Kang, Department of Biostatistics, University of Michigan, Ann Arbor, MI, USA.
Email: jiankang@umich.edu
Tianwei Yu, School of Data Science and Warshel Institute, The Chinese University of Hong Kong - Shenzhen, and Shenzhen Research Institute of Big Data, Shenzhen, China.
Email: yutianwei@cuhk.edu.cn

Funding information
National Institutes of Health, Grant/Award Numbers: R01GM124061, R01HL095479, R01MH105561

Abstract

Jointly analyzing transcriptomic data and the existing biological networks can yield more robust and informative feature selection results, as well as better understanding of the biological mechanisms. Selecting and classifying node features over genome-scale networks has become increasingly important in genomic biology and genomic medicine. Existing methods have some critical drawbacks. The first is they do not allow flexible modeling of different subtypes of selected nodes. The second is they ignore nodes with missing values, very likely to increase bias in estimation. To address these limitations, we propose a general modeling framework for Bayesian node classification (BNC) with missing values. A new prior model is developed for the class indicators incorporating the network structure. For posterior computation, we resort to the Swendsen-Wang algorithm for efficiently updating class indicators. BNC can naturally handle missing values in the Bayesian modeling framework, which improves the node classification accuracy and reduces the bias in estimating gene effects. We demonstrate the advantages of our methods via extensive simulation studies and the analysis of the cutaneous melanoma dataset from The Cancer Genome Atlas.

KEYWORDS

Bayesian nonparametrics, false discovery rate control, feature selection, gene networks

1 | INTRODUCTION

Feature selection is a fundamental problem in high-dimensional data analysis. Existing biological networks, including biological pathways and molecular interactions, have been found to be helpful for depicting the biological relationship between the features. In the field of transcriptomics, each node in the biological network corresponds to a feature measured in the high-dimensional data. Researchers are interested in classifying the network node features in different categories according to their biological characteristics and behavior in the transcriptomics data. We refer to this procedure as node classification on the network.

Node classification is different from the traditional differential expression framework which calculates false discovery rates, that is, posterior probabilities of differential expression using parametric or nonparametric density estimations, without considering biological relations between features.[1,2]

For the classification of network nodes into "selected" and "unselected" categories, some filtering algorithms were developed in the machine learning and bioinformatics fields, without much consideration of statistical inference.[3-6] In the statistics field, the main approach for network-based feature selection is built under the parametric/regression framework,

such as References 7-19, where model structures are developed to capture the dependency of genes by using various penalties that smooth the regression coefficients of the features over the network, or applying different priors utilizing the structure of the network.

Bayesian hierarchical models have been developed to cluster genes in complex high-dimensional data.[20-22] We have previously developed the Bayesian network feature finder (BANFF),[23] a Bayesian nonparametric method for selection of network nodes.[24] Unlike the regression-based methods, BANFF allows any type of association between features and outcome variables, or even testing behavior of the features without an outcome variable. It achieves this goal by conducting the analysis in two steps. First, a test statistic is generated for each node in a univariate analysis, which can be supervised if an outcome variable is involved in the analysis. Second, unsupervised node classification is conducted using a Bayesian nonparametric approach that takes into account both the dependency structure on the network and the test statistic of each node.

While BANFF is suitable for analyzing data where the association of nodes with an outcome variable is considered, it can also handle some situations that existing regression-based approaches cannot. Some examples include: (1) there is no outcome variable, and some intrinsic properties of the nodes are summarized into a test statistic;[24] (2) the study design is complex and case-control type regression methods are not suitable, such as those involving longitudinal or functional measurements. However, BANFF still has some major limitations. First, it assumes the test statistics of the null distribution follow a symmetric distribution; while in many applications the null distribution of test statistics appear to be asymmetric. Second, it lumps up-regulated and down-regulated genes into a single group and assumes they behave symmetrically. Third, it does not handle missing values in a systematic modeling approach, thus it may lose power to detect signals while increasing the false positive rates. Fourth, the posterior computation algorithm may suffer slow convergence in some applications.

In this work, we address these issues by developing a more flexible framework for Bayesian node classification (BNC). BNC allows an asymmetric null distribution, as well as different levels of deviation from the null, for example, different degree of deviation for down-regulated and up-regulated genes. Different from BANFF, BNC adopts more efficient posterior computation algorithm, the Swendsen-Wang algorithm, and it can naturally handle missing values.

Missing data is an important issue in the network-based gene expression analysis. Here by missingness we refer to the lack of observation on certain nodes across all samples, which is a common situation in gene expression data.[25] laid the foundation of missing mechanisms and provided ideas on how to handle missingness. However, most of the approaches do not handle the missing of entire rows in the data well. In our motivating dataset, among the genes with at least one connection, around 6% are not measured, and another 14% are observed but of a low and unreliable expression level for statistical testing. We treat them as nodes with missing observations in the network. As a result, the occurrence of missingness makes our problem even more challenging. Ignoring such nodes and their edges, as all existing methods do, causes severe loss of network structure and biases the results. Thus handling missing nodes in the network is of great importance to our problem, due to the fact that missing nodes are possible to be either down-regulated or up-regulated genes, and/or serve to communicate information via their edges with observed genes (Figure 1).

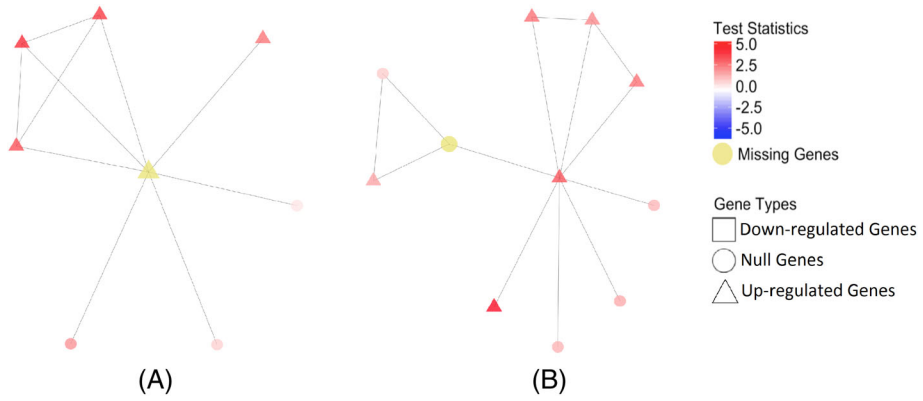

FIGURE 1 The impact of missing gene nodes in the network. (A) The missing gene is itself an up-regulated gene, it would be excluded if missing genes are removed from data analysis; (B) the missing gene serves as a "bridge" for information exchange. If it is simply removed, the light red node located on the left side would not be able to be recalled as up-regulated gene

Our proposed BNC is a nonparametric Bayesian method without imposing any parametric assumptions on the distributions of the test statistics for each class. Instead, we use the Dirichlet process mixture (DPM) model. DPM is widely used and extensively studied from the literature (see Reference 26 for an overview of DPM). To specify the prior for the network node specific class indicators, BNC adopts a weighted Potts prior, which generalized the Ising prior from two categories to multi-categories that can satisfy the three-class feature classification problem. Our proposed BNC can be seen as an extension of the local false discovery rate control rule proposed by Efron et al[27] to adopt the extra information of the network and classify network nodes. We developed an `R` package `BNC` (https://github.com/kangjian2016/BNC) to implement the proposed method.

The remainder of the manuscript is organized as follows. In Section 2, we describe the proposed model and the prior specifications. In Section 3, we present the posterior computation algorithms. In Section 4, we compare the performance of the proposed method with the traditional methods via extensive simulation studies. In Section 5, we analyze the cutaneous melanoma dataset and discuss biologically meaningful results.

2 | BAYESIAN NODE CLASSIFICATION

2.1 | The model

Consider a network consisting of n nodes. At each node $i(i = 1, \dots, n)$, we obtain a node-specific test statistic, denoted r_i. Let $\mathbf{C} = \{c_{ij}\}$ be the adjacency matrix characterizing the gene network configuration, where $c_{ij} = 1$ if genes i and j are biologically connected and $c_{ij} = 0$ otherwise. In gene expression differentiation analysis, each node represents a gene, and r_i is obtained for testing gene behaviors. There are three common gene behaviors: "down-regulated," "up-regulated," and "not differentiated expressed," to which we refer as the "null genes." Let $z_i \in \{-1, 1, 0\}$ indicate the latent class for node i and values -1, 1, and 0 represent "down-regulated," "up-regulated," and "null" genes, respectively. We consider a Bayesian nonparametric model:

$$[r_i|(\mu_i, \sigma_i^2)] \sim \mathrm{N}(\mu_i, \sigma_i^2), \tag{1}$$

$$[(\mu_i, \sigma_i^2)|z_i] \sim G_{-1}I(z_i = -1) + G_0 I(z_i = 0) + G_1 I(z_i = 1), \tag{2}$$

$$G_k \sim \mathcal{DP}(G_{0k}, \tau_k), \tag{3}$$

where $G_k(k = -1, 0, 1)$ is a random probability measure defined on $\mathbb{R} \times [0, \infty)$ following the Dirichlet process with base measure G_{0k} and precision parameter τ_k. The domain of G_{0k} is the same as G_k. We choose the conjugate priors, that is, $(\mu, \sigma^2) \sim G_{0k}$ is equivalent to $\mu|\sigma^2 \sim \mathrm{N}(\mu_{0k}, \sigma^2\phi_{0k})$ and $\sigma^2 \sim \mathrm{IG}(a_{0k}, b_{0k})$.

To incorporate this topology structure, we assign a weighted Potts prior to $\mathbf{z} = (z_1, \dots, z_n)$, denoted by $\mathrm{wPotts}(\boldsymbol{\pi}, \boldsymbol{\rho}, \mathbf{w}, \mathbf{C})$, where $\boldsymbol{\pi} = (\pi_{-1}, \pi_0, \pi_1)$ with $\pi_k > 0$, $\boldsymbol{\rho} = (\rho_{-1}, \rho_0, \rho_1)$ with $\rho_k \geq 0$ and $\mathbf{w} = (w_1, \dots, w_n)$ with $w_i \geq 0$. Then the probability mass function is proportional to

$$\exp\left[\sum_{i=1}^{n}(\tilde{\omega}_i \log(\pi_{z_i}) + \rho_{z_i}\sum_{i\neq j}\omega_j c_{ij} I[z_i = z_j])\right]. \tag{4}$$

The parameter $\boldsymbol{\pi}$ contains prior knowledge about the distribution of the class indicator $\mathbf{z}$. We assume that $\pi_1 + \pi_{-1} < \pi_0$ implying that signals are sparse. Similar to the Ising model, parameter ρ_k controls the global strength of the neighborhood similarity. When $\rho_k = 0, z_i$ is independent with z_j for j in the neighborhood of i. However, when $\rho_k > 0, z_i$ has a larger probability to take the value of k when $z_j = k$ for j in the neighborhood of i. Across the whole gene network, the larger the ρ_k is, the stronger the tendency of genes to share the same memberships with neighbors. Weight w_i can be elicited from the prior biological knowledge. A larger weight w_i implies a stronger prior belief of the similarity between gene i and its neighbors locally. The neighbor weight $\widetilde{w}_i = \sum_{j=1}^{n} c_{ij}w_j / \sum_{i=1}^{n} c_{ij}$ represents the average of weights from neighbors for gene i.

2.2 | Missing data problem and model representation

Our goal is to make inference on the latent class z_i from the observed network node-specific test statistics. However, as the test statistics are not always fully observed in real data analysis, we introduce a missing data indicator s_i for gene i, $s_i = 1$ if r_i is missing, $s_i = 0$ if r_i is observed. The missing test statistics introduce great challenges in classifying the features, thus,

we utilize gene network information to help in classifying gene nodes. For gene i, the objective is to make inference about z_i when its test statistics r_i, missing indicator s_i and gene network information are provided. We assume the distribution of missing test statistics is the same with the distribution of the test statistics been observed. To be specific, given gene i in the class of k, that is, $z_i = k$, we further introduce a cluster index g_i of gene i, g_i represents the cluster index indicating which component in the mixture model that r_i is associated with. In particular, r_i given g_i is assumed to be normally distributed with mean $\tilde{\mu}_{g_i}$ and variance $\tilde{\sigma}^2_{g_i}$, denoted by $N(\tilde{\mu}_{g_i}, \tilde{\sigma}^2_{g_i})$. We write $\widetilde{\theta}_g = (\tilde{\mu}_{g_i}, \tilde{\sigma}^2_{g_i})$ and assume they are independently drawn from a base measure called G_{0k}. The $\tilde{\boldsymbol{\theta}}$ denotes all the $\tilde{\theta}_g$s for simplicity. Given $z_i = k$, g_i follows a discrete distribution with parameter $\mathbf{a}_k, \mathbf{q}_k$, which means g_i can take values in $\boldsymbol{a_k} = (a^k_1, a^k_2, \dots, a^k_{L_k})$ with probability $\mathbf{q}_k = (q^k_1, q^k_2, \dots, q^k_{L_k})$, denoted as Discrete$(\boldsymbol{a}_k, \boldsymbol{q}_k)$. In fact, the actual values of g_i given $z_i = k$ can be arbitrary as long as they can be differentiated from each other, thus, we assume $\mathbf{a}_k = (1, 2, \dots, L_k)$ without loss of generality. The probability $\mathbf{q}_k$ follows a Dirichlet distribution with parameters $(\tau_k/L_k, \tau_k/L_k, \dots, \tau_k/L_k)$. Note that the total number of components L_k for all $k = -1, 0, 1$ are unknown, this extended DPM model is nonparametric in nature. In summary, we have the following Bayesian hierarchical model:

$$\begin{aligned} r_i|g_i, s_i = 0, \tilde{\boldsymbol{\theta}} &\sim N(\tilde{\mu}_{g_i}, \tilde{\sigma}^2_{g_i}), \\ g_i|z_i = k, \boldsymbol{q}_k &\sim \text{Discrete}(\boldsymbol{a}_k, \boldsymbol{q}_k), \\ \widetilde{\boldsymbol{\theta}}_g &\sim G_{0k} \quad \text{for } g \in \boldsymbol{a}_k, \\ \boldsymbol{q}_k &\sim \text{Dirichlet}(\tau_k \mathbf{1}_{L_k}/L_k), \\ \mathbf{z} &\sim \text{wPotts}(\boldsymbol{\pi}, \boldsymbol{\rho}, \mathbf{w}, \mathbf{C}) \end{aligned} \tag{5}$$

where test statistics $\{r_i : s_i = 0\}$ and the network configuration $\mathbf{C}$ are observed data. The latent class $\mathbf{z}$ is of our primary interest for Bayesian inference.

2.3 | Methods for handling missing data

When the test statistics $\mathbf{r}$ are partially observed, the nodes with missing $\mathbf{r}$ values can still serve to pass information between their neighboring nodes. More importantly, some nodes with missing $\mathbf{r}$ values can still belong to the significant classes, and their neighboring nodes with observed $\mathbf{r}$ values can provide evidence. In order to infer the class labels of nodes with missing $\mathbf{r}$ values, we conduct inference on the missing $\mathbf{r}$ values of such nodes. The test statistics $\mathbf{r}$ can be partitioned into two parts $\mathbf{r} = (\mathbf{r}_{mis}, \mathbf{r}_{obs})$ with $\mathbf{r}_{mis} = \{r_i : s_i = 1\}$ and $\mathbf{r}_{obs} = \{r_i : s_i = 0\}$. Similarly, we can also partition the cluster indices into the observed component and the missing component as $\mathbf{g} = (\mathbf{g}_{mis}, \mathbf{g}_{obs})$. The element-wise representation of the missing component of the test statistics is $\mathbf{r}_{mis} = (r_{mis,1}, \dots, r_{mis,m})$ and the cluster indices are $\mathbf{g}_{mis} = (g_{mis,1}, \dots, g_{mis,m})$ where m is the number of missing nodes in the network.

Under the fully Bayesian inference framework, the missing values are one type of latent variables in the model. We can make posterior inference on the joint distribution of $\mathbf{r}_{mis}$ and all the other latent quantities in the model. From the model representation (5), test statistics are conditionally independent given their cluster indices and density specifications, which means the conditional distribution for $\mathbf{r}_{mis}$ given $\mathbf{r}_{obs}, \mathbf{g}, \mathbf{z}, \tilde{\boldsymbol{\theta}}$ only depends on $\mathbf{g}_{mis}, \tilde{\boldsymbol{\theta}}$:

$$P(\mathbf{r}_{mis}|\mathbf{r}_{obs}, \mathbf{g}_{obs}, \mathbf{g}_{mis}, \mathbf{z}_{obs}, \mathbf{z}_{mis}, \tilde{\boldsymbol{\theta}}) = P(\mathbf{r}_{mis}|\mathbf{g}_{mis}, \tilde{\boldsymbol{\theta}}) = \Pi^m_{i=1} P(r_{mis,i}|g_{mis,i}, \tilde{\boldsymbol{\theta}})$$

This further implies that in the posterior computation algorithm (see Section 3) when there are missing gene nodes in the network, we only need to introduce one more step to impute the missing test statistics $r_{mis,i}, i = 1, \dots, m$ within each iteration. Assume the superscript represents the results from the previous iteration tth, for the $(t+1)$th iteration, we only need to draw a imputed value for $r^{(t+1)}_{mis,i}$ from $N(\tilde{\mu}^{(t)}_{g_{mis,i}}, \tilde{\sigma}^{2(t)}_{g_{mis,i}})$.

We also propose a fast imputation approach by approximating the fully Bayesian inference based on the assumption that neighboring genes are more likely to share the same functionalities. We can integrate out all the latent quantities in the model and impute $r_{mis,i}$ using $\mathbf{r}_{obs}$ based on the conditional expectation:

$$E(r_{mis,i}|\mathbf{r}_{obs}) = \int r_{mis,i} P(r_{mis,i}|\mathbf{r}_{obs}) dr_{mis,i}$$

where

$$P(r_{mis,i}|\mathbf{r}_{obs}) = \int\int P(r_{mis,i}|z_{mis,i},\tilde{\boldsymbol{\theta}})P(z_{mis,i},\widetilde{\boldsymbol{\theta}|\mathbf{r}_{obs}})\widetilde{dz_{mis,i}}d\tilde{\boldsymbol{\theta}}.$$

Suppose we have N samples of $(z_{mis,i},\tilde{\boldsymbol{\theta}})$ from the posterior distribution given $\mathbf{r}_{obs}$, denoted as $(z^{(1)}_{mis,i},\boldsymbol{\theta}^{(1)}),\ldots,(\widetilde{z^{(N)}_{mis,i}},\widetilde{\boldsymbol{\theta}^{(N)}})$, then $P(r_{mis,i}|\mathbf{r}_{obs})$ can be approximated by $N^{-1}\sum_{n=1}^{N}P(r_{mis,i}|z^{(n)}_{mis,i},\widetilde{\boldsymbol{\theta}^{(n)}})$.

As indicated by model (4), when $\rho > 0, z_{mis,i}$ has a larger probability to take the value of k when $z_j = k$ for j in the neighborhood of i. From our experience, we can approximate $P(r_{mis,i}|\mathbf{r}_{obs})$ by a discrete distribution $P(r_{mis,i} = r_j|\mathbf{r}_{obs}) = 1/|nbr(i)|$ for $j \in nbr(i)$, where $nbr(i)$ represents the neighborhood of i with r_j observed. Then $E(r_{mis,i}|\mathbf{r}_{obs})$ can be approximated by $\sum_{j\in nbr(i)} r_j/|nbr(i)|$ which is exactly the average of neighboring observed test statistics. We refer to this approach as the nearest-neighbor imputation method.

3 | POSTERIOR COMPUTATION

The posterior computation algorithm has three major steps in each iteration: (1) Impute missing test statistics $\mathbf{r}_{mis}$ (if any) either by conditional sampling (fully Bayesian inference) or by the nearest-neighbor imputation method; (2) Update class indicators $\mathbf{z}$ by the Swendsen-Wang algorithm, and (3) Update $\tilde{\boldsymbol{\theta}}$ by refitting a DPM to estimate densities for each regulation type. Others including L_k, g_k are omitted temporarily for simplicity. For updating the hyperparameters in the Potts model for $\mathbf{z}$, we adopt the method of Double Metropolis-Hastings (DMH) sampler proposed by Liang.[28]

3.1 | Swendsen-Wang algorithm

It has been widely used in the Potts model. It works by introducing another set of auxiliary variables denoted as $\mathbf{W} = \{W_{ij}, i \sim j\}$. W_{ij} is defined only when gene pairs i and j are connected. Given z_i, z_j, W_{ij} is uniformly distributed between 0 and $\exp(\rho_{z_i}\omega_j c_{ij} I[z_i = z_j])$. Then the full conditional distribution for $\mathbf{z}$ given $\mathbf{W}$ can be simplified as proportional to $P(\mathbf{r}|\mathbf{z},\tilde{\boldsymbol{\theta}})\exp\left[\sum_{i=1}^{n}\tilde{\omega}_i\log(\pi_{z_i})\right]$.

The posterior sampling scheme has two steps: the network partitioning step (sample $\mathbf{W}$ given $\mathbf{z}$) and the network relabeling step (sample $\mathbf{z}$ given $\mathbf{W}$). The objective for network partitioning is to cut the network into smaller connected subnetworks so that the genes located within the same subnetwork share the same class indicators. Then in the network relabeling step, the class indicators of all the genes located within the same subnetwork can be flipped simultaneously. Comparing to the Gibbs sampler when it updates the genes each one at a time, the Swendsen-Wang algorithm advantages itself by a more efficient group level updating scheme and a better convergence.

3.2 | DPM density updating

Conditional on the class indicators, we update g_i and $\tilde{\theta}_i$ given $g_1,\ldots,g_{i-1},\tilde{\theta}_1,\ldots,\tilde{\theta}_{i-1}$. Utilizing algorithm 8 in Reference 29, we first summarize the frequency for each of the total l unique g values ever appeared in set $(g_1,\ldots,g_{i-1})$, denoted as $(1,2,\ldots,l)$ with cluster parameters $(\tilde{\theta}_1,\ldots,\tilde{\theta}_l)$. It is $n_{i,g} = \sum_{j=1}^{i-1} I[g_j = g], g = 1,2,\ldots,l$. Then the prior probability of g_i equals to any of the ever-appeared cluster index g is given by $n_{i,g}/(i-1+\tau_k), g \in (1,2,\ldots,l)$, if the sampled g_i equals to any appeared cluster index g, then we set $\tilde{\theta}_i = \tilde{\theta}_g$; on the other hand, the prior probability of g_i being a new index is given by $\tau_k/(i-1+\tau_k), g \notin (1,2,\ldots,l)$, if the sampled g_i is a new index, then we sample a new set of parameter $\tilde{\theta}_g$ from base measure G_{0k}. Given the cluster index g, r_i follows a normal distribution with parameter $\tilde{\theta}_g$. In every iteration, we maintain the order of $G_k(k = -1,0,1)$ by swapping the labels if necessary.

3.3 | Choice of initial values

In order to speed up the convergence in Markov chain Monte Carlo, we specify the initial values for $G_{0k}, (k = -1,0,1)$, $\mathbf{z},\mathbf{g},\tilde{\theta}$, and $\mathbf{L}$ based on the DPM density fitting of the test statistics $\boldsymbol{r}$ without the network information, we develop the Kullback-Leibler-divergence-based hierarchical ordered density clustering algorithm (KL-HODC). In the beginning, we

TABLE 1 Simulation settings

	Down-regulated class	Null class	Up-regulated class
Gaussian	N(−0.6, 0.2)	N(0, 0.2)	N(0.6, 0.2)
Gamma	Gamma (shape=2, scale=0.5)	N(0, 0.4)	Gamma (shape=2, scale=0.3)
	Truncated within $(-\infty, 2]$, shifted −1.9		Truncated within $(-\infty, 1.8]$, shifted +1.7
Log-normal	log N(0, 1)	N(0, 0.4)	log N(0, 1)+2.2
	Truncated within $(-\infty, 2]$, shifted −1.9		Truncated within $(-\infty, 2.3]$, shifted +2.2

order all the small cluster density parameters $\tilde{\theta}_g, \tilde{\theta}_g = (\widetilde{\mu}_g, \widetilde{\sigma}_g^2)$ based on their mean value $\widetilde{\mu}_g$ locations. Each time, we pick several clusters to form a proposed null. We calculate the KL distance between this proposed null and a prior null which is pre-determined by biological knowledge. The combination of the clusters with the smallest KL distance is selected and added as the initial value for the null densities.

Once all the clusters are assigned to three classes, $\mathbf{z}, \mathbf{g}, \tilde{\theta}, L$ can be determined as well. When the biology knowledge is not available for the prior null, it can be estimated by a truncated bi-Gaussian distribution using the central part of the test statistics such as statistics within 15% and 75% quantiles.

KL-HODC is a hierarchical density clustering algorithm that substantially improves the existing algorithm HODC.[24] The KL-HODC incorporates the biological knowledge as a prior null density and it is able to handle the multi-class feature classification problem, while HODC can only be used for selecting features, not further differentiating their subtypes.

4 | SIMULATION STUDIES

We conduct extensive simulation studies to evaluate the performance of the proposed methods for the complete data case and the missing data case.

4.1 | Settings

The network used in the simulation studies is a subnetwork of the real biological network used in real data analysis downloaded from the High-quality INTeractomes (HINT) database.[30] It is formed by a total of 776 nodes with a median degree of 3, a mean degree of 5.2 and a maximum degree of 30. The underlying true gene regulation types are assigned based on the merged communities by the fast greedy modularity optimization algorithm.[31] We assign the genes located in the largest community as the null class and then we randomly assign the down-regulated or the up-regulated class to the other two. For the null genes, their test statistics are independently drawn from a normal distribution, and for the up-regulated or the down-regulated genes, their test statistics are independently drawn from one of the following three distributions: a normal, a gamma, or a lognormal (see Figure 2 for an illustration of one simulated dataset; see Table 1 for the designs of the simulation settings). The missing locations are randomly selected among the genes with network degrees less than 6, which is the 66% quantile of the degrees of the nodes in the network. We simulate 20% missingness since it is the missing rate in the real dataset.

4.2 | Evaluation criteria

For each simulation setting, we simulate 50 datasets in total, indexed by $s = 1, \ldots, 50$. A classification rate of the genes with true class indicator $z_i = a$ being classified as b for $a, b = -1, 0, 1$ is defined by $\sum_{s=1}^{50} I[\hat{z}_i^{(s)} = b, z_i = a]/50$, where $\hat{z}_i^{(s)}$ is the estimate of z_i in the simulated dataset s. Denote TP-down, TP-up and TN the true positive rate averaged across all simulations for the down-regulated ($a = b = -1$), up-regulated ($a = b = 1$), and null genes ($a = b = 0$), respectively. Denote FN-down and FN-up the averaged false negative rates for the down-regulated and up-regulated genes. Additionally, FP-down and FP-up are the averaged false positive rates into the down-regulated and up-regulated classes, respectively.

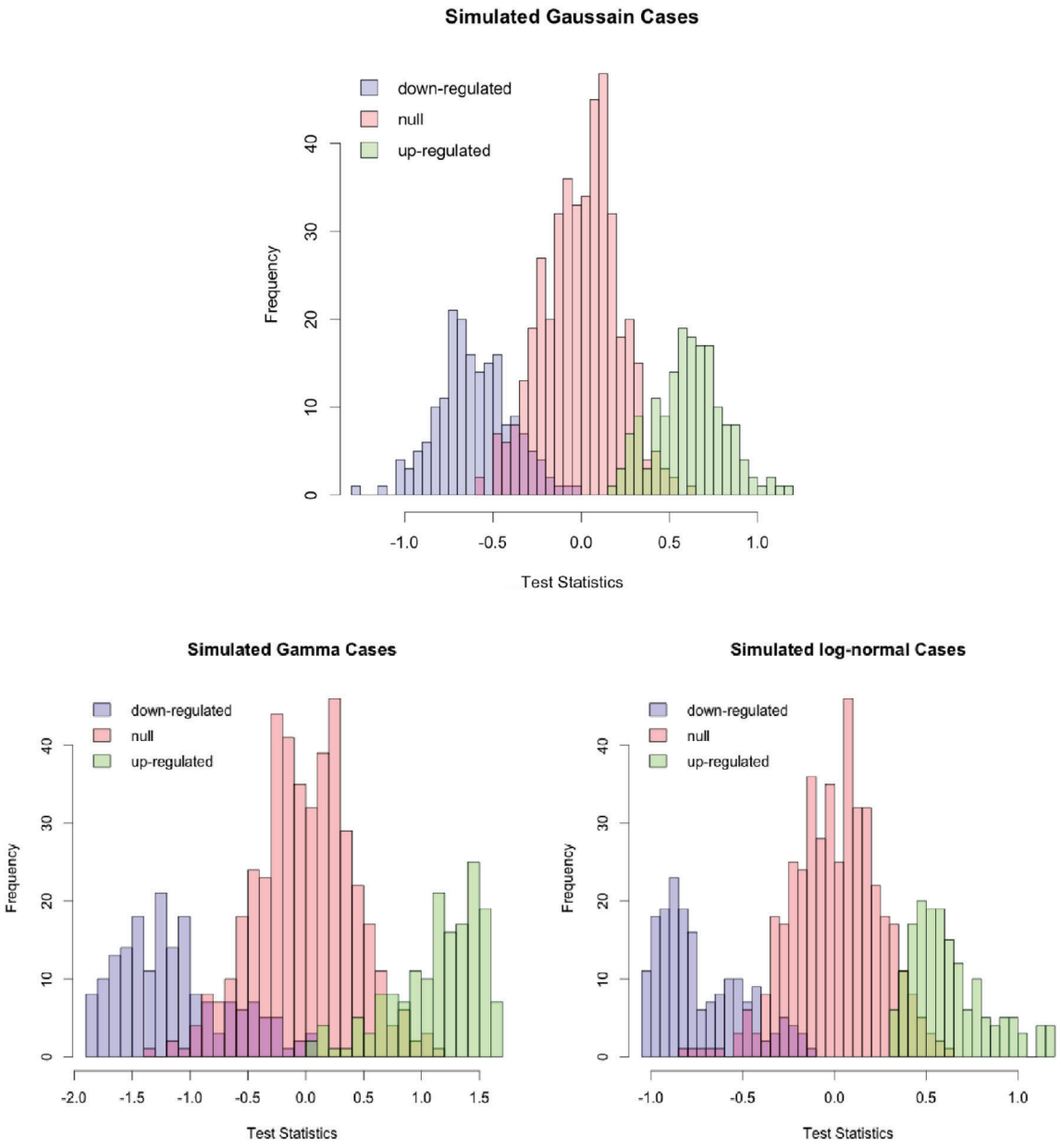

FIGURE 2 An illustration of the distributions of test statistics under each simulation setting

And finally, FDR is the false discovery rate defined as the proportion of false discoveries among all the discoveries on average.

4.3 | Hyperprior specifications

For hyperprior settings in the Potts prior model (4), we set weights $\omega_j = 1, j = 1, 2, \ldots, n$. Then $\tilde{w}_i = 1, i = 1, 2, \ldots, n$. We set $\boldsymbol{\rho} = (1.001, 0.497, 0.998)$, $\boldsymbol{\pi} = (0.15, 0.70, 0.15)$ as an output from DMH of a 10 000-iteration run with 5000 burn-ins. The proposal used in DMH for $(\boldsymbol{\pi}, \boldsymbol{\rho})$ is an independent random walk proposal for $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$: for each element of $\boldsymbol{\rho}$, it is a truncated Gaussian distribution with a mean of 0, a SD of 0.03, a lower-bound of 0, and a upper-bound of 1.5; for $\boldsymbol{\pi}$, since it must satisfy $\pi_2 = 1 - \pi_1 - \pi_3$ and $\pi_1 + \pi_3 < 0.5$, thus we assume π_1 and π_3 follow the truncated Gaussian distribution with a mean of 0, a SD of 0.03, a lower-bound of 0 and a upper-bound of 0.5. As for the hyperparameters for the DPM

model fitting, For each regulation type k, we assume the base measure $G_{0k} = P(\mu, \Sigma)$ is conjugate Normal-inverse-Gamma (NIWG) distribution with parameters $(\mu_{0k}, c_{0k}, S_{0k}, \psi_{0k})$ and the scale parameter c_{0k} in the normal part of the base measure follows a gamma distribution with parameters (a_{0k}, b_{0k}). In general, we denote the distribution G_{0k} as NIWG$(\mu_{0k}, S_{0k}, \psi_{0k}, a_{0k}, b_{0k})$. For this prior model, we first apply the normal mixture modeling for model-based clustering method (Mclust by Fraley and Raftery[32]) where the parameter indicating the total number of groups is set to be 3. Then we use the estimated mean and variance from each group k as μ_{0k} and S_{0k}. And we set $\phi_{0k} = 3, a_{0k} = 1, b_{0k} = 100, \tau_k = 3$.

4.4 | Simulation results for the complete observed case

We first consider the cases when all the test statistics are fully observed. For each of these simulation settings, we compare our method (BNC) with the Bayesian nonparametric mixture model for selecting genes (BANFF) by Zhao et al.[24] and the false discovery rate controlling procedures for identifying differentially expressed genes (locfdr) by Efron.[1] The locfdr method does not consider the network structure and only uses the gene-level test statistics.

BANFF is a Bayesian nonparametric gene and gene-network selection method, it can also utilize the network information but it is mainly for selecting the activated-state genes from the null genes. In order to modify BANFF for this feature classification problem, we first classify genes into three groups by MCLUST—Gaussian finite mixture models fitted via EM algorithm.[32] Then we flip the sign of the test statistics of the genes assigned to the down-regulated class so that ideally those genes combined with the up-regulated class should be of the active state. Then the finalized class indicators are assigned based on the results from BANFF being flipped back. For the locfdr, it is a kernel density-based non-parametric method for selecting differentially expressed genes without considering the network. To be specific, we applied the central matching for estimating the null densities and then calculated the estimated local false discovery rate for each gene. We adopt a commonly used cutoff of 0.2 so that the genes with the posterior probability of being in the null class below 0.2 will be identified as differentially expressed, and the null class otherwise. Then the differentially expressed genes can be further classified by comparing the relative locations of their test statistics with 0.

Table 2 indicates that for Gaussian simulations, under each regulation type, BNC performs better than the BANFF and locfdr. BNC achieves classification accuracies as high as 0.87 for the down-regulated genes, 0.91 for the up-regulated genes, and 0.97 for the null genes. At the same time, BNC achieves the false positive rates as lower as 0 for the down-regulated genes, 0.03 for the up-regulated genes, 0.12 for the null genes to be classified as down-regulated genes, 0.09 for the null genes to be classified as up-regulated genes. Overall, BNC can achieve higher accuracies and lower false positive and false negative rates. BANFF performs worse in the true negative rates and false positive rates. locfdr performs well at selecting the null genes, with a true negative rate of 1. However, it gives a false negative rate as high as 0.49 for the down-regulated genes and 0.5 for the up-regulated genes, indicating the procedure is overly conservative.

Comparing the classification accuracies for Gamma and log-normal settings, BNC outperforms all the others in all the measures. The BANFF performs worse than the BNC. It is because the proposed method can flexibly model the gene subtypes so that it allows for different levels of deviation from the null for down-regulated and up-regulated genes.

TABLE 2 Algorithm performance for complete data cases

Generative model	Methods	TP-down	TP-up	TN	FP-down	FP-up	FN-down	FN-up	FDR
Gaussian	BNC	0.87	0.91	0.97	0	0.03	0.12	0.09	0.03
	BANFF	0.75	0.87	0.62	0.03	0.36	0.2	0.13	0.3
	locfdr	0.5	0.51	1	0	0	0.49	0.5	0.01
Gamma	BNC	0.92	0.96	0.99	0	0.01	0.08	0.04	0.01
	BANFF	0.5	0.89	0.69	0	0.31	0.38	0.11	0.2
	locfdr	0.57	0.71	0.98	0.01	0.01	0.43	0.29	0.03
Log-normal	BNC	0.9	0.96	0.99	0	0.01	0.1	0.04	0.01
	BANFF	0.73	0.92	0.55	0.05	0.04	0.17	0.08	0.31
	locfdr	0.59	0.72	0.99	0.01	0.01	0.41	0.28	0.03

The worse performance of locfdr compared to BNC indicates that by utilizing network information, better classification accuracies can be obtained.

4.5 | Simulation results for the missing data case

We further compare our method with the others when there are missing node observations in the network. We only focus on the symmetric cases as described in Table 1, and compare five combinations of methods to perform feature classification and to handle missingness simultaneously: (1) BNC+Bayes: we apply the BNC for feature classification and the conditional sampling for fully Bayesian inference to impute the missing test statistics. (2) BNC+NN: we apply the BNC for feature classification combined with the nearest neighbor imputation method to impute the missing test statistics. (3) BNC+NArm: we first remove all the missing nodes and their edges in the network and then use BNC for feature classification. In this case, only the estimated class indicators for gene nodes with observed test statistics can be obtained. (4) BANFF+NN: we utilize the BANFF for feature classification and use the nearest neighbor imputation method to impute the missing test statistics. (5) BANFF+NArm: we apply BANFF to the reduced network comprised of nodes with observed test statistics. Similar to BNC+NArm, only replace the BNC with BANFF for feature classification.

To summarize the classification accuracies, we separate different types of nodes to calculate the averaged rates: (1) Missing: only average the rates among the genes whose test statistics are missing. (2) Observed: only average the rates among all observed genes. (3) Total: average among all the genes nodes.

From Table 3, we observe that BNC+NN performs the best in general. The overall classification accuracies for the all the down-regulated, the up-regulated and the null genes to be correctly classified are 0.87, 0.87, 0.89. The averaged false positive rates for the null genes being classified as down-regulated or up-regulated are 0 and 0.01. The averaged false negative rates for the down-regulated or the up-regulated genes are 0.12 and 0.13, respectively. The estimated false discovery rate is 0.12. This performance keeps consistent among missing genes and the observed genes. Compared to BNC+Bayes, BNC+NN is slightly better. It is because the nearest-neighbor imputation scheme is more flexible than the model-based Bayesian posterior inferences since Bayesian posterior sampling needs to specify a proper prior. The Bayesian model we are utilizing might not characterize very well the predictive distribution of the missing test statistics given the observed test statistics across the network while utilizing the information from the nearest neighbors might help to improve.

The accuracies will drop if we use the BANFF for feature classification regardless of which schemes are used for handling the missingness. It indicates that our proposed algorithm outperforms BANFF when there are missing observations in the network, which is consistent with the simulation results in fully observed cases. Moreover, regardless of which feature classification algorithms we utilize, either BNC or BANFF, comparing NArm

TABLE 3 Algorithm performance for missing data cases

Algorithm	Gene nodes type	TP-down	TP-up	TN	FP-down	FP-up	FN-down	FN-up	FDR
BNC+Bayes	Missing	0.73	0.75	0.77	0.01	0.22	0.26	0.25	0.27
	Observed	0.92	0.9	0.78	0	0.22	0.05	0.1	0.2
	Total	0.88	0.88	0.78	0.01	0.22	0.1	0.12	0.21
BNC+NN	Missing	0.83	0.81	0.88	0.01	0.11	0.17	0.19	0.15
	Observed	0.88	0.88	0.89	0	0.01	0.11	0.12	0.11
	Total	0.87	0.87	0.89	0	0.01	0.12	0.13	0.12
BNC+NArm	Observed	0.87	0.88	0.66	0.01	0.33	0.07	0.12	0.3
BANFF+NN	Missing	0.6	0.79	0.48	0.04	0.48	0.26	0.21	0.47
	Observed	0.7	0.86	0.41	0.05	0.54	0.19	0.14	0.46
	Total	0.68	0.85	0.42	0.05	0.53	0.21	0.15	0.46
BANFF+NArm	Observed	0.67	0.82	0.5	0.05	0.45	0.23	0.18	0.43

with the imputation methods Bayes or NN among the observed gene nodes, we observe that the classification accuracies drop and the false positive/false negative rates increase, and so does the averaged false discovered rates. Thus, imputation methods are recommended for feature classification problem with missing gene observations.

5 | SURVIVAL ANALYSIS OF CUTANEOUS MELANOMA

5.1 | Dataset

We analyze the cutaneous melanoma dataset from The Cancer Genome Atlas (TCGA),[33] downloaded from the cBio Cancer Genomics Portal.[34] There are 478 patient records by the time we downloaded. After removing six patient records that lack gene expression profiles, one patient record with a negative survival month due to possible errors, one patient record that is missing survival status, and one patient record that is missing the sample type which is one of the covariates we are interested in, we use the remaining 469 patient records in a Cox proportional hazard model to assess the association between the expression levels of individual genes and survival time. In our model, we control for three confounders: age at initial pathologic diagnosis (minimum 15, median 58, mean 58.08, max 90, and 8 are missing), gender (180 females and 290 males), and sample type (366 of metastatic, 102 of primary tumor, and 1 of additional metastatic).

We downloaded the protein-protein interactions in Homo sapiens from the High-quality INTeractomes (HINT) database by Das and Yu.[30] After data cleaning, there are a total of 11 662 genes and 87 482 edges. Then we apply the community detection algorithm by Clauset et al[31] to extract the largest connected subnetwork as our network input. To be specific, the largest connected component contains 10 484 genes while the remaining genes form 1097 tiny islands (1 island is of five genes, 2 islands are of four genes, 5 are of three genes, 61 of two genes, and 1028 are formed by a single gene node). By excluding these tiny islands, the network contains a total of 10 484 nodes, with a degree distribution of a minimum of 1, a median of 3, a mean of 8.328, and a maximum of 400.

For the gene expression profile, we first map all 20 530 unique gene names to 18 978 Entrez IDs. Among the 10 484 genes in the network, 9833 can be mapped to an expression profile. There are 651 (6.21%) genes that do not have any expression profile and another 1433 (13.67%) genes that are considered unreliably measured based on their low maximum expression level across the samples. Removing such genes leads to a total missing rate of 19.88% in our real data analysis. For each gene included in the analysis, we first fitted the Cox proportional hazard model while controlling for the three confounders. The z-statistic for the gene was extracted from the model and used in the node classification analysis.

Similar to the simulations, for the Potts prior model, the hyperparameters in Equation (4), we prefix the $\omega_j = 1, j = 1, 2, \ldots, n$ so that the $\tilde{w}_i = 1, i = 1, 2, \ldots, n$. Set $\boldsymbol{\rho} = (1.003, 0.479, 0.988)$ and $\boldsymbol{\pi} = (0.15, 0.70, 0.15)$ as an output from the DMH of 10 000 iterations with 5000 burn-in. Other hyperparameters settings are the same with the settings used for simulations. In the following discussion, we refer to genes that significantly increase the risk of death as high-risk genes, and genes that significantly decrease the risk of death as low-risk genes.

5.2 | Results

Our method finds 144 high-risk genes and 263 low-risk genes. Compared to ours, the locfdr method finds 217 low-risk genes by central matching estimation for a symmetric null while it does not identify any differentially expressed genes by applying a split normal version of central matching estimation for an asymmetric null. Thus, for the following discussion, we will focus on the comparison between the proposed method and the locfdr utilizing central matching estimation (see Figure 3A) even though the null density is asymmetric and the mode of the distribution is away from zero for the motivating dataset.

Using the test statistics alone, combined with the common assumption of symmetric null distribution, locfdr identifies significant genes only on the low-risk side (see Figure 3B). On the other hand, when the existing network is utilized, the proposed method can detect both high-risk and low-risk genes.

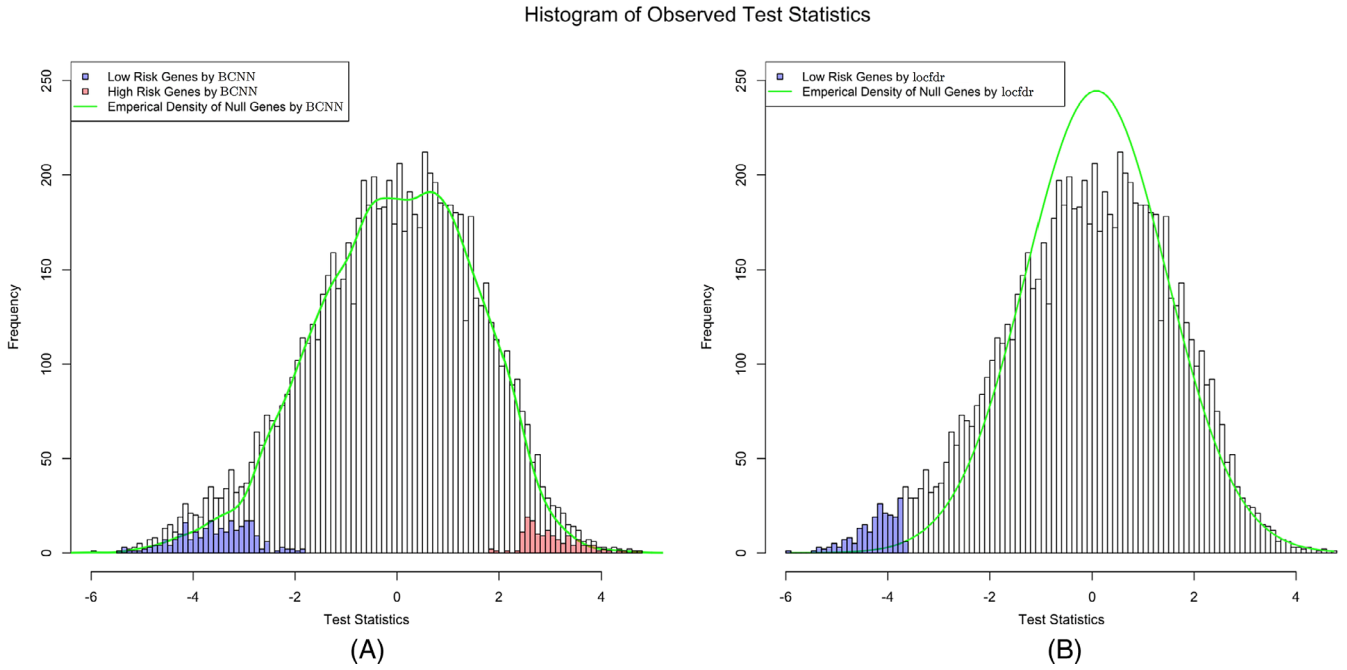

FIGURE 3 Histogram of the test statistics, with estimated null density and frequencies of the selected genes. (A) Results by BNC; (B) results by locfdr with center matching estimation for a symmetric null. Local false discovery rate is controlled at 0.2 for both methods. Blue, low-risk genes; red, high-risk genes

To facilitate interpretation, we further find modules by applying the fast greedy community detection algorithm among the selected nodes and their one-step neighbors.[31] There is a total of 56 modules selected, 16 of which contain more than 10 selected genes.

Here we present some example modules and discuss their biological functions in relation to the clinical outcome. The module shown in Figure 4A contains 48 selected genes. There are 39 low-risk genes and 9 high-risk genes in this module. Analyzing the biological functions of the selected genes using GOstats,[35] we find the biological function of the low-risk genes are focused in the area of immune responses, with 18 of the 39 genes falling into the biological process of "regulation of immune response," and various related functions. The prognosis of melanoma is closely related to tumor-infiltrating lymphocytes.[36] A cross-platform meta-analysis has shown that the increased expression of immune function-related genes in melanoma is associated with longer patient survival, and B and T cells are enriched in melanoma biopsies from patients with favorable outcome.[37] The low-risk genes selected from this bulk RNAseq data likely represent higher level of immune cell infiltration in patients with better survival outcome.

The module shown in Figure 4B contains 23 high-risk genes and 17 low-risk genes. An interesting finding is that the top gene ontology biological process being over-represented by the high-risk genes is transmembrane transport, with eight of the 23 genes falling into this category. Six of the high-risk genes are involved in ion transport. Although transmembrane transporters have not been systematically studied in melanoma progression, recent developments in other cancer have indicated their role in cancer prognosis.[38] For example, among the genes selected by BNC, gene 3764 (KCNJ8) encodes a potassium channel. It is found to be over-expressed in nasopharyngeal carcinoma (NPC) tissues as well as in esophageal cancer.[39,40] The gene 6520 (SLC3A2) encodes the heavy chain of the transmembrane protein CD98 that regulates intracellular calcium levels and transports L-type amino acids. It has been linked to Ras-driven skin carcinogenesis and prognosis of lung cancer.[41,42] Gene 11 660 (ABCC9) is a member of the ATP-binding cassette transporter (ABC transporter) family. Recently the down-regulation of ABC transporters, including ABCC9, has been observed in prostate cancer.[43] Gene 255 738 (PCSK9) is involved in peptide precursors trafficking. It has been shown that tumor development influences the host lipid metabolism through PCSK9-mediated degradation of hepatic LDLR, and PCSK9 is suppressed in hepatocellular carcinoma.[44,45] Combined with these evidence in other types of cancer, our results indicate a link between transmembrane transporters and the prognosis of melanoma.

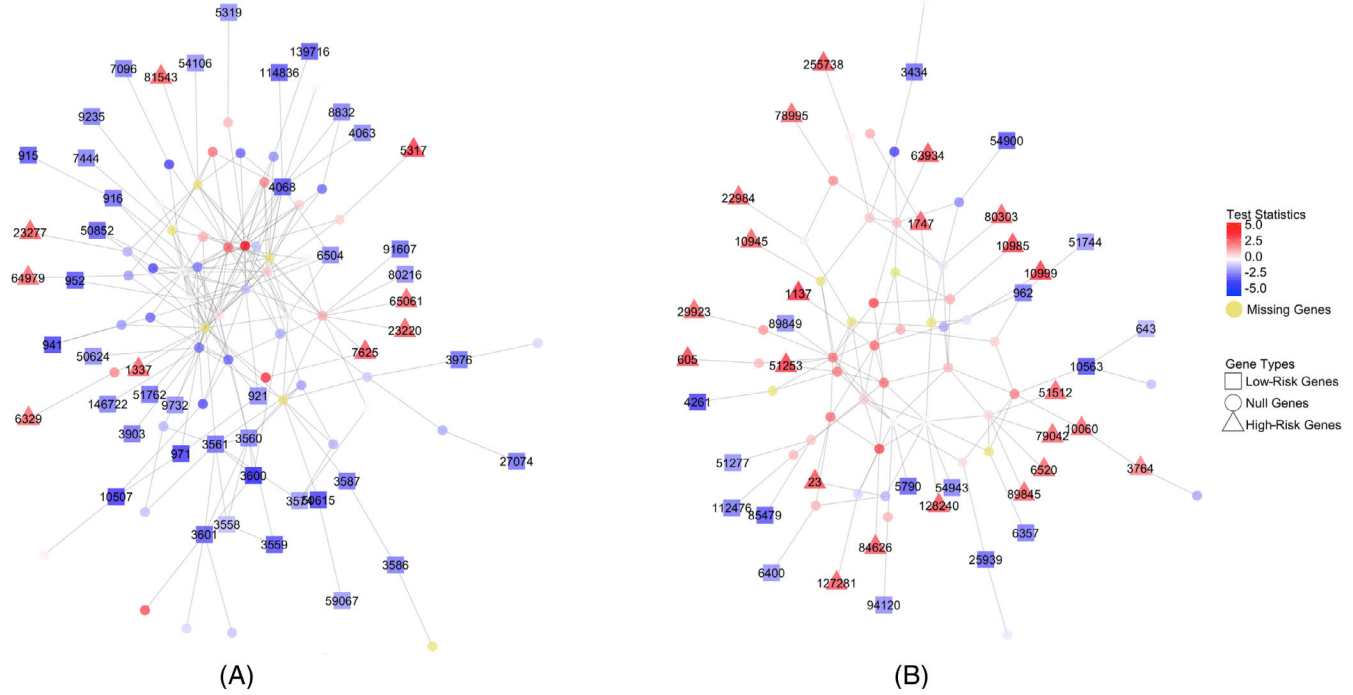

FIGURE 4 Two example modules of selected genes. (A) An example module with 39 low-risk genes and 9 high-risk genes; (B) An example module with 23 high-risk genes and 17 low-risk genes

Six of the 17 low-risk genes belong to cytokine-mediated signaling pathways, which are critical in leukocyte trafficking and immune functions.[46] Gene 643 (CXCR5), a member of the CXC chemokine receptor family, is expressed in mature B-cells and Burkitt's lymphoma. The loss of CXCR5 in naive T cells is linked to the metastatic dissemination of melanoma into lungs.[47] Gene 3434 (IFIT1) is an interferon-induced protein. Overexpression of IFIT1 has been shown to predict improved outcome in newly diagnosed glioblastoma.[48] Gene 4261 (CIITA) regulates class II major histocompatibility complex gene transcription. CIITA overexpression facilitates engulfment of the T-cell material by melanoma cells, which can blunt the anti-tumor response.[49] Gene 10 563 (CXCL13) is a cytokine that belongs to the CXC chemokine family. Its expression is correlated with the densities of tumor high endothelial venules (HEVs), which allows the recruitment of tumor-infiltrating lymphocytes (TILs).[50] CXCL13 is also found to be one of a group of diagnostic markers of melanoma.[51] Gene 25 939 (SAMHD1) is a deoxyribonucleoside triphosphate triphosphohydrolase that decreases dNTP pools, which in turn affects DNA replication fidelity. Although it has not been well studied in melanoma, SAMHD1 is found to be frequently mutated in colon cancers, resulting in decreased SAMHD1 activity and thereby facilitating cancer cell proliferation.[52]

Figure 5 shows a module where two nodes with missing observations are identified as low-risk genes. These two genes, 3135 (HLA-G) and 3133 (HLA-E) have both been implicated in melanoma immunomodulation. HLA-G can inhibit the function of T cells, natural killer (NK) cells, and dendritic cells. It has been documented that HLA-G is inconsistently expressed in melanoma, and its expression can provide the malignant cells a mechanism of escaping immune surveillance.[53,54] Similarly, HLA-E expression on the cell surface facilitates the melanoma cells' escape from CTL and NK cell surveillance.[55] Among all the 13 genes in this module, 10 are annotated to the biological process of regulation of immune response, which is consistent with our earlier discussion about the association of immune function-related genes with patient survival.[36,37] The figure also shows that by test statistic alone, three of the 13 genes are not selected by locfdr. They are selected by BNC because their connections in the network offer extra evidence that they are related to the clinical outcome. These three genes are 910 (CD1B), 3811 (KIR3DL1), 3823 (KLRC3). It has been found that down-regulating CD1 molecules including CD1B on infiltrating dendritic cells by secreting IL-10 are associated with metastasis of melanoma.[56] Both KIR3DL1 and KLRC3 are receptors expressed on NK cells, the induction of which shows the potential of suppressing solid melanoma tumors.[57]

Besides being biologically relevant, the selected modules each has good predictive power on the clinical outcome. Here we compare concordance statistics which is commonly used in survival analysis to check on model validity. Concordance statistics (C-statistics) is defined as the probability of agreement between any two randomly chosen observations. If a

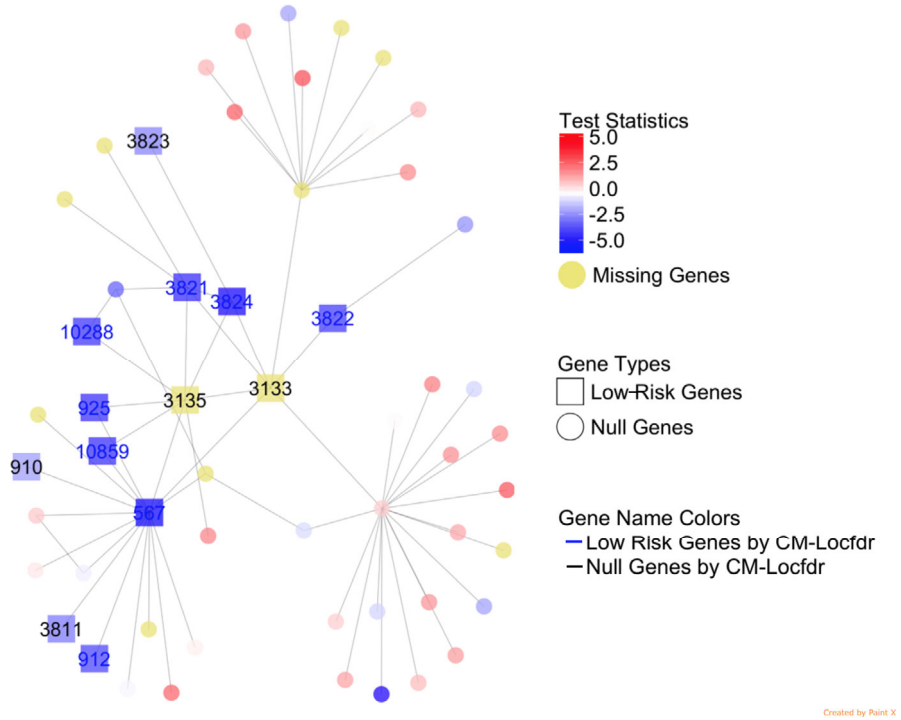

FIGURE 5 A module containing two nodes with missing observations being identified as low-risk genes by BNC

TABLE 4 Module group sizes and concordant scores

Module ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Total number of genes nodes	116	101	86	61	57	40	40	37	28	27	26	26	24	21	21	20
High-risk genes by BNC	24	9	23	7	9	10	7	1	3	2	4	8	6	4	2	0
Low-risk genes by BNC	29	39	17	24	15	12	11	16	11	10	8	8	8	5	8	11
Low-risk genes by locfdr	13	16	7	8	11	5	6	8	7	7	4	3	1	2	0	7
Low-risk genes by both methods	12	13	6	8	9	5	5	8	7	7	3	3	1	2	0	7
C-statistics by BNC	0.7491	0.7363	0.7311	0.7113	0.7196	0.7146	0.7097	0.6846	0.6788	0.6902	0.688	0.6996	0.6928	0.7006	0.6695	0.6806
C-statistics by locfdr	0.6702	0.6504	0.6648	0.6899	0.6761	0.6635	0.6688	0.649	0.6497	0.661	0.6558	0.6679	0.6557	0.6612	NA	0.6667

model predicts a higher risk of death of one patient when it is observed with a shorter survival time compared to the other, then we define this pair as "agree," otherwise as "disagree." Since ties of the predicted and the observed survival time may occur, we refer to those pairs are "tie." Then, the C-statistics is defined as $P(agreement) = (agree + tied/2)/(agree + disagree + tied)$ for all possible comparable pairs.[58] By saying "comparable," it is defined as the opposite to "uncomparable." The "uncomparable" pairs are the pairs when we lack the information of whether the predicted and the survival time agree or disagree with each other. For example, one patent record is censored at time 2 while the survival time we

predict is 4. In general a C-statistic of 1 means perfect agreement; 0.6-0.7 is a common result for survival data while 0.5 is an agreement that is no better than the random guess.

We then calculate the C-statistics by the direct comparisons between the observed survival time and the predicted survival time generated by the model fitting results of the Cox proportional hazard model for each selected module. Due to the lack of the ability to handle the nodes when their expression profiles are completely missing in the Cox proportional hazard model, thus, all the models are fitted using data except for those missing nodes. The modules with the number of genes larger than 20 are outputted in Table 4. From the Table 4, we observe that our proposed method can successfully recall the high-risk genes when they cannot be discovered by locfdr method. The averaged C-statistics for these top 16 modules are 0.70 for our method while it is 0.66 for locfdr. This indicates a better predicting power using our method.

6 | DISCUSSION

The feature classification problem utilizing existing network information is a novel problem which has drawn increasing attention recently. Based on our knowledge, we are the first to propose a non-parametric Bayesian framework not only to select features but also to differentiate the subtypes of the selected features over genome-scale networks, and to handle the missing node observations simultaneously.

We have applied our method to the cutaneous melanoma dataset from TCGA. The results provided novel gene regulation evidence for unveiling the disease mechanism. In general, we recommend BNC for feature classification over the network. If there are missing node observations in the network, we recommend nearest-neighbor imputation method to handle missingness.

It is noteworthy that in the application section, we do not consider genes that are not part of the network because the main purpose of the subsequent analysis is to select subnetworks, which are functionally coherent and easy to interpret.

Moreover, the KL-HODC algorithm we proposed for setting up the initial values for fast convergence can be further utilized in another fast version of our proposed algorithm based on density approximations, which can be implemented in our package. The fast algorithm works by fitting DPM densities for several iterations and then the densities are fixed, the algorithm continues to run but only update the class indicators given the densities until the Markov Chain reaches its equilibrium. For this fast version, it is of great importance to choose an initial value based on our experience. Thus, KL-HODC advantages itself by providing a better inference of the density specifications and class indicators since it can properly incorporate the prior biological knowledge.

Future work may include the extension of our method to a multivariate statistics cases when combined information can provide more aspect of the information for classifying features, which can intuitively improve the classification accuracy. In addition, much of the gene network is directional, including signal transduction and regulatory relations. In the current work, we assumed the network was non-directional. There have been some related work that allow for the structure of a directed graph under the regression framework.[59,60] Given the complexities due to the structure of loops, substantial modification of our model is necessary to adapt it to directed graphs.

ACKNOWLEDGEMENT

This study was partially funded by NIH Grants R01MH105561, R01HL095479, and R01GM124061.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the cBio cancer genomics portal at http://www.cbioportal.org/, reference number.[34]

ORCID

Tianwei Yu https://orcid.org/0000-0003-2502-1628

REFERENCES

1. Efron B, Tibshirani R. Empirical Bayes methods and false discovery rates for microarrays. *Genet Epidemiol*. 2002;23:70-86.
2. Do KA, Müller P, Tang F. A Bayesian mixture model for differential gene expression. *J Royal Stat Soc Ser C (Appl Stat)*. 2005;54:627-644.
3. Cun YP, Fröhlich H. Biomarker gene signature discovery integrating network knowledge. *Biology*. 2012;1:5-17.
4. Cun YP, Fröhlich H. Network and data integration for biomarker signature discovery via network smoothed T-statistics. *PLoS One*. 2013;8:e73074.

5. Apolloni J, Leguizamón G, Alba E. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Appl. Soft Comput*. 2016;38:922-932.
6. Kong Y, Yu T. A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data. *Bioinformatics*. 2018;34(21):3727-3737. doi:10.1093/bioinformatics/bty429
7. Wei Z, Li HZ. A Markov random field model for network-based analysis of genomic data. *Bioinformatics*. 2007;23:1537-1544.
8. Li CY, Li HZ. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*. 2008;24:1175-1182.
9. Pan W, Xie BH, Shen XT. Incorporating predictor network in penalized regression with application to microarray data. *Biometrics*. 2010;66:474-484.
10. Wei P, Pan W. Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model. *Bioinformatics*. 2008;24(3):404-411. doi:10.1093/bioinformatics/btm612
11. Wei P, Pan W. Network-based genomic discovery: application and comparison of Markov random-field models. *J Royal Stat Soc Ser C-Appl Stat*. 2010;59:105-125. doi:10.1111/j.1467-9876.2009.00686.x
12. Li F, Zhang NR. Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *J Am Stat Assoc*. 2012;105(491):1202-1214.
13. Stingo FC, Vannucci M. Variable selection for discriminant analysis with Markov random field priors for the analysis of microarray data. *Bioinformatics*. 2011;27:495-501.
14. Stingo FC, Chen YA, Tadesse MG, Vannucci M. Incorporating biological information into linear models: a Bayesian approach to the selection of pathways and genes. *Ann Appl Stat*. 2011;5(3):1978-2002.
15. Ročková V, George EI. EMVS: the EM approach to Bayesian variable selection. *J Am Stat Assoc*. 2014;109:828-846.
16. Sun H, Lin W, Feng R, Li H. Network-regularized high-dimensional cox regression for analysis of genomic data. *Stat Sin*. 2014;24(3):1433-1459. doi:10.5705/ss.2012.317
17. Dona MSI, Prendergast LA, Mathivanan S, Keerthikumar S, Salim A. Powerful differential expression analysis incorporating network topology for next-generation sequencing data. *Bioinformatics*. 2017;33(10):1505-1513. doi:10.1093/bioinformatics/btw833
18. Ren J, Du Y, Li S, Ma S, Jiang Y, Wu C. Robust network-based regularization and variable selection for high-dimensional genomic data in cancer prognosis. *Genet Epidemiol*. 2019;43(3):276-291. doi:10.1002/gepi.22194
19. Zhang W, Ota T, Shridhar V, Chien J, Wu B, Kuang R. Network-based survival analysis reveals subnetwork signatures for predicting outcomes of ovarian cancer treatment. *PLoS Comput Biol*. 2013;9(3):e1002975. doi:10.1371/journal.pcbi.1002975
20. Wei P, Pan W. Bayesian joint modeling of multiple gene networks and diverse genomic data to identify target genes of a transcription factor. *Ann Appl Stat*. 2012;6(1):334-355. doi:10.1214/11-Aoas502
21. Wu Z, Casciola-Rosen L, Rosen A, Zeger SL. A Bayesian approach to restricted latent class models for scientifically structured clustering of multivariate binary outcomes. *Biometrics*. 2020. doi:10.1111/biom.13388
22. Zhou F, He K, Li Q, Chapkin RS, Ni Y. Bayesian biclustering for microbial metagenomic sequencing data via multinomial matrix factorization. *Biostatistics*. 2021. doi:10.1093/biostatistics/kxab002
23. Lan Z, Zhao Y, Kang J, Yu T. Bayesian network feature finder (BANFF): an R package for gene network feature selection. *Bioinformatics*. 2016;32(23):3685-3687.
24. Zhao Y, Kang J, Yu TW. A Bayesian nonparametric mixture model for selecting genes and gene subnetworks. *Ann Appl Stat*. 2014;8:999.
25. Little RJA, Rubin DB. *Statistical Analysis with Missing Data*. Hoboken, NJ: John Wiley & Sons; 2014.
26. Neal RM. Markov Chain sampling methods for Dirichlet process mixture models. *J Comput Graph Stat*. 2012;9(2):249-265.
27. Efron B, Storey JD, Tibshirani R. Microarrays empirical Bayes methods, and false discovery rates. 2001; Stanford University, Department of Statistics.
28. Liang F. A double Metropolis–Hastings sampler for spatial models with intractable normalizing constants. *J Stat Comput Simul*. 2010;80:1007-1022.
29. Neal RM. Markov chain sampling methods for Dirichlet process mixture models. *J Comput Graph Stat*. 2000;9:249-265.
30. Das J, Yu HY. HINT: high-quality protein interactomes and their applications in understanding human disease. *BMC Syst Biol*. 2012;6:92.
31. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. *Phys Rev E*. 2004;70:066111.
32. Fraley C, Raftery AE. MCLUST: software for model-based cluster analysis. *J Classif*. 1999;16:297-306.
33. Network CGA. Genomic classification of cutaneous melanoma. *Cell*. 2015;161:1681-1696.
34. Cerami E, Gao JJ, Dogrusoz U, et al. The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discov*. 2012;2:401-404.
35. Falcon S, Gentleman R. Using GOstats to test gene lists for GO term association. *Bioinformatics*. 2007;23:257-258.
36. Taylor RC, Patel A, Panageas KS, Busam KJ, Brady MS. Tumor-infiltrating lymphocytes predict sentinel lymph node positivity in patients with cutaneous melanoma. *J Clin Oncol Offic J Am Soc Clin Oncol*. 2007;25:869-875.
37. Lardone RD, Plaisier SB, Navarrete MS, et al. Cross-platform comparison of independent datasets identifies an immune signature associated with improved survival in metastatic melanoma. *Oncotarget*. 2016;7:14415-14428.
38. Elsnerova K, Mohelnikova-Duchonova B, Cerovska E, et al. Gene expression of membrane transporters: Importance for prognosis and progression of ovarian carcinoma. *Oncol Rep*. 2016;35:2159-2170.
39. Zhou W, Feng XL, Li H, et al. Functional evidence for a nasopharyngeal carcinoma-related gene BCAT1 located at 12p12. *Oncol Res*. 2007;16:405-413.

40. Warnecke-Eberz U, Metzger R, Hölscher AH, Drebber U, Bollschweiler E. Diagnostic marker signature for esophageal cancer from transcriptome analysis. *Tumour Biol J Int Soc Oncodevelop Biol Med*. 2016;37:6349-6358.
41. Guo X, Li HW, Fei F, et al. Genetic variations in SLC3A2/CD98 gene as prognosis predictors in non-small cell lung cancer. *Mol Carcinog*. 2015;54(Suppl 1):E52-E60.
42. Estrach S, Lee SA, Boulter E, et al. CD98hc (SLC3A2) loss protects against ras-driven tumorigenesis by modulating integrin-mediated mechanotransduction. *Cancer Res*. 2014;74:6878-6889.
43. Demidenko R, Razanauskas D, Daniunaite K, Lazutka JR, Jankevicius F, Jarmalaite S. Frequent down-regulation of ABC transporter genes in prostate cancer. *BMC Cancer*. 2015;15:683.
44. Bhat M, Skill N, Marcus V, et al. Decreased PCSK9 expression in human hepatocellular carcinoma. *BMC Gastroenterol*. 2015;15:176.
45. Huang JF, Li L, Lian JH, et al. Tumor-induced hyperlipidemia contributes to tumor growth. *Cell Rep*. 2016;15:336-348.
46. Zlotnik A, Yoshie O. The chemokine superfamily revisited. *Immunity*. 2012;36:705-716.
47. Jacquelot N, Enot DP, Flament C, et al. Chemokine receptor patterns in lymphocytes mirror metastatic spreading in melanoma. *J Clin Investigat*. 2016;126:921-937.
48. Zhang JF, Chen Y, Lin GS, et al. High IFIT1 expression predicts improved clinical outcome, and IFIT1 along with MGMT more accurately predicts prognosis in newly diagnosed glioblastoma. *Human Pathol*. 2016;52:136-144.
49. Lloyd MC, Szekeres K, Brown JS, Blanck G. Class II transactivator expression in melanoma cells facilitates T-cell engulfment. *Anticancer Res*. 2015;35:25-29.
50. Martinet L, Le Guellec S, Filleron T, et al. High endothelial venules (HEVs) in human melanoma lesions: major gateways for tumor-infiltrating lymphocytes. *Oncoimmunology*. 2012;1:829-839.
51. Liu WT, Peng YH, Tobin DJ. A new 12-gene diagnostic biomarker signature of melanoma revealed by integrated microarray analysis. *PeerJ*. 2013;1:e49.
52. Rentoft M, Lindell K, Tran P, et al. Heterozygous colon cancer-associated mutations of SAMHD1 have functional significance. *Proc Nat Acad Sci U S A*. 2016;113:4723-4728.
53. Paul P, Rouas-Freiss N, Khalil-Daher I, et al. HLA-G expression in melanoma: a way for tumor cells to escape from immunosurveillance. *Proc Nat Acad Sci U S A*. 1998;95:4510-4515.
54. Yan WH, Lin AF, Chang CC, Ferrone S. Induction of HLA-G expression in a melanoma cell line OCM-1A following the treatment with 5-aza-2'-deoxycytidine. *Cell Res*. 2005;15:523-531.
55. Derré L, Corvaisier M, Charreau B, et al. Expression and release of HLA-E by melanoma cells and melanocytes: potential impact on the response of cytotoxic effector cells. *J Immunol*. 2006;177:3100-3107.
56. Gerlini G, Tun-Kyi A, Dudli C, Burg G, Pimpinelli N, Nestle FO. Metastatic melanoma secreted IL-10 down-regulates CD1 molecules on dendritic cells in metastatic tumor lesions. *Am J Pathol*. 2004;165:1853-1863.
57. Wennerberg E, Kremer V, Childs R, Lundqvist A. CXCL10-induced migration of adoptively transferred human natural killer cells toward solid tumors causes regression of tumor growth in vivo. *Cancer Immunol Immunother CII*. 2015;64:225-235.
58. Therneau T A package for survival analysis in S. version 2.38; 2015.
59. Ni Y, Stingo F, Baladandayuthapani V. Bayesian graphical regression. *J Am Stat Assoc*. 2019;114:184-197.
60. Ni Y, Muller P, Wei L, Ji Y. Bayesian graphical models for computational network biology. *BMC Bioinform*. 2018;19:63.

How to cite this article: Jin Z, Kang J, Yu T. Feature selection and classification over the network with missing node observations. *Statistics in Medicine*. 2022;41(7):1242-1262. doi: 10.1002/sim.9267

APPENDIX A

Equation derivations

Swendsen-Wang Suppose $\mathbf{W} = \{W_{ij}, i \sim j\}$ where the W_{ij} is defined only when gene pair i and j are connected. The distribution of W_{ij} is

$$P(W_{ij}|z_i, z_j) = \exp(-\rho_{z_i}\omega_j C_{ij} I[z_i = z_j]) \times I[0 \le W_{ij} \le \exp(\rho_{z_i}\omega_j C_{ij} I[z_i = z_j])]$$

Then the conditional distribution of $\mathbf{W}$ given $\mathbf{z}$ is:

$$P(\mathbf{W}|\mathbf{z}) \propto \exp\left(\sum_{i=1}\sum_{j\neq i} -\rho_{z_i}\omega_j c_{ij} I[z_i = z_j]\right) \prod_{i=1}\prod_{j\neq i} I[0 \le W_{ij} \le \exp(\rho_{z_i}\omega_j c_{ij} I[z_i = z_j])]$$

Algorithm 1. Fully Bayesian posterior updating algorithm

Input observed test statistics $\mathbf{r} = (\mathbf{r}_{obs}, \mathbf{r}_{mis})$, adjacency matrix $\mathbf{C} = \{c_{ij}\}$, τ, $\mathbf{w}$, π=NULL, ρ=NULL, ρ_0, $\mathbf{r}_0$, $\mathbf{z}$, PriorNullDensity=NULL, PriorForDPMDensityFitting, ParaForMCMC, rhoSD, rhoUpperBound, rhoLowerBound, piSD, piUpperBound, piLowerBound, MissingDataImputationMethod, TotalNumIterationsForDMH, nSaveForDMH, TotalNumIterations, nSave

Initialization:

if (is.null(PriorNullDensity)) **then**

PriorNullDensity ← BiGaussianDensityByCentralFitting($\mathbf{r}_{obs}$)

end if

$(\mathbf{z}, \mathbf{g}, \widetilde{\theta}, \mathbf{L})$ ← KL-HODC($\mathbf{r}_{obs}$, PriorForDPMDensityFitting, ParaForMCMC)

if (is.null(π) | is.null(ρ)) **then**

(π, ρ) ← DMH($\mathbf{C}$, $\mathbf{r}_{obs}$, ρ_0, $\mathbf{r}_0$, $\mathbf{z}$, rhoSD, rhoUpperBound, rhoLowerBound, piSD, piUpperBound, piLowerBound, TotalNumIterationsForDMH, nSaveForDMH)

end if

$\mathbf{r}_{mis}$ ← Mean($\mathbf{r}_{obs}$)

Loop:

zTrace ← $\mathbf{z}$

Iter ← 0

while (Iter < TotalNumIterations) **do**

$\mathbf{z}$ ← SW($\mathbf{C}, \mathbf{z}, \mathbf{r}_{obs}, \widetilde{\theta}, \rho, \pi$)

$(\widetilde{\theta}, \mathbf{g})$ ← DPMDensityFitting($\mathbf{C}, \mathbf{z}, \mathbf{r}$, PriorForDPMDensityFitting, ParaForMCMC)

$\mathbf{r}_{mis}$ ← MissingDataImputation(MissingDataImputationMethod, $\mathbf{C}, \mathbf{r}, \mathbf{g}, \widetilde{\theta}$)

zTrace ← cbind(zTrace, $\mathbf{z}$)

Iter ← Iter+1

end while

ClassIndicators ← ClassIndicatorsWithLocalFDRControl(zTrace, nSave)

return ClassIndicators

Algorithm 2. Function: prior null density fitted as bi-Gaussian density

function BiGaussianDensityByCentralFitting($\mathbf{r}$, QuantileForFitting=NULL)

if is.null(QuantileForFitting) **then**

QuantileForFitting ← $c(0.25, 0.75)$

end if

CentralTestStat ← $\mathbf{r}$[which($\mathbf{r} \in$ QuantileForFitting)]

CutOff ← quantile($\mathbf{r}$,0.5)

NormalFitForUpRegulateClass ←

NormalDensityFitting(CentralTestStat>CutoffWithItsReflected)

NormalFitForDownRegulateClass ←

NormalDensityFitting(CentralTestStat<CutoffWithItsReflected)

return CutOff, NormalDensityForUpRegulateClass, NormalDensityForDownRegulateClass

end function

Algorithm 3. Function: initial values based on KL-HODC

function KL-HODC(**r**, PriorForDPMDensityFitting, ParaForMCMC, PriorNullDensity)
  $(\mathbf{g}, \widetilde{\theta}) \leftarrow$ DPdensity(**r**, PriorForDPMDensityFitting, ParaForMCMC)
  $(\mathbf{g}, \widetilde{\theta}) \leftarrow$ SortClusterByMeanLocation($\mathbf{g}, \widetilde{\theta}$)
  procedure (initialize null class index)
    $D_{min} \leftarrow +\infty$
    NullClassIndex $\leftarrow \emptyset$
    DownRegulateClassIndex $\leftarrow \emptyset$
    UpRegulateClassIndex $\leftarrow \emptyset$
    for all $l_0 \in s$ **do**
      CandidateNullDensity $\leftarrow \{\widetilde{\theta}_{l_0}\}$
      D $\leftarrow$ KLDistance(CandidateNullDensity, PriorNullDensity)
      if $D < D_{min}$ **then**
        $D_{min} \leftarrow D$
        NullClassIndex $\leftarrow \{l_0\}$
        DownRegulateClassIndex $\leftarrow \{l'\}_{\forall l', 1 \le l' < l_0}$
        UpRegulateClassIndex $\leftarrow \{l'\}_{\forall l', l' > l_0}$;
      end if
    end for
  end procedure
  procedure (merge multiple clusters to search for clusters in null class)
    $D_{\text{diff}} \leftarrow +\infty$
    while $D_{\text{diff}} > 0$ & DownRegulateClassIndex $\neq \emptyset$ & UpRegulateClassIndex $\neq \emptyset$ **do**
      CandidateNullClass $\leftarrow$ NullClassIndex $\cup \{l_0 + 1\}$
      CandidateNullDensity $\leftarrow$ CandidateNullDensity $\cup \{\widetilde{\theta}_{l_0+1}\}$
      $D_+ \leftarrow$ KLDistance(CandidateNullDensity, PriorNullDensity)
      CandidateNullClass $\leftarrow$ NullClassIndex $\cup \{l_0 - 1\}$
      CandidateNullDensity $\leftarrow$ CandidateNullDensity $\cup \{\widetilde{\theta}_{l_0+1}\}$
      $D_- \leftarrow$ KLDistance(CandidateNullDensity, PriorNullDensity)
      if $D_- \le D_+$ **then**
        NullClassIndex $\leftarrow$ NullClassIndex $\cup \{l_0 - 1\}$
        DownRegulateClassIndex $\leftarrow$ DownRegulateClassIndex$\{l'\}_{\forall l', 1 \le l' < (l_0 - 1)}$
        $D_{\text{diff}} \leftarrow D_{min} - D_-$
        $D_{min} = D_-$
      else
        NullClassIndex $\leftarrow$ NullClassIndex $\cup \{l_0 + 1\}$
        UpRegulateClassIndex $\leftarrow$ UpRegulateClassIndex$\{l'\}_{\forall l', l' > (l_0 + 1)}$;
        $D_{\text{diff}} \leftarrow D_{min} - D_+$
        $D_{min} = D_+$
      end if
    end while
  end procedure
  $\mathbf{z} \leftarrow \mathbf{z} = (z_1, \ldots, z_n), \forall i \in$ NullClassIndex, $z_i = 0, \forall i \in$ DownRegulateClassIndex, $z_i = -1, \forall i \in$ UpRegulateClassIndex, $z_i = +1$
  $\mathbf{g} \leftarrow \mathbf{z}$
  $\widetilde{\theta} \leftarrow \widetilde{\theta} = \{\widetilde{\theta}_{g_i}\}$
  $\mathbf{L} \leftarrow c(|\text{DownRegulateClassIndex}|, |\text{NullClassIndex}|, |\text{UpRegulateClassIndex}|)$
  return $\mathbf{z}, \mathbf{g}, \widetilde{\theta}, \mathbf{L}$
end function

Algorithm 4. Function: hyperparameters by Double Metropolis-Hasting

function DMH(Network, TestStat, $\boldsymbol{\rho}$, $\mathbf{r}$, $\mathbf{z}$, rhoSD, rhoUpperBound, rhoLowerBound, piSD, piUpperBound, piLowerBound, TotalNumIterations, nSave)
  rhoTrace $\leftarrow \boldsymbol{\rho}$
  piTrace $\leftarrow \mathbf{r}$
  Iter $\leftarrow 0$
  for (Iter < TotalNumIterations) **do**
    repeat
      $\boldsymbol{\rho}' = (\boldsymbol{\rho}'_1, \boldsymbol{\rho}'_2, \boldsymbol{\rho}'_3, \boldsymbol{\rho}'_4) \leftarrow$ rtruncnorm(1, $\boldsymbol{\rho}$, rhoSD, rhoLowerBound, rhoUpperBound)
      $\boldsymbol{\pi}' = (\boldsymbol{\pi}'_1, \boldsymbol{\pi}'_2, \boldsymbol{\pi}'_3) \leftarrow$ rtruncnorm(1, $\boldsymbol{\pi}$, rhoSD, rhoLowerBound, rhoUpperBound)
      $\boldsymbol{\pi}'_2 \leftarrow 1 - \boldsymbol{\pi}'_1 - \boldsymbol{\pi}'_3$
    until $\boldsymbol{\rho}'_1 > \boldsymbol{\rho}'_2$ & $\boldsymbol{\rho}'_3 > \boldsymbol{\rho}'_2$ & $\boldsymbol{\pi}'_2 > 0.5$
    $\mathbf{z}' \leftarrow$ DrawSampleFromPriorModel(Network, TestStat, $\boldsymbol{\rho}', \boldsymbol{\pi}'$)
    LogAcceptRate $\leftarrow$ LogDataLikelihood(Network, TestStat, $\mathbf{z}', \boldsymbol{\rho}, \boldsymbol{\pi}$) + LogDataLikelihood(Network, TestStat, $\mathbf{z}, \boldsymbol{\rho}', \boldsymbol{\pi}'$) -LogDataLikelihood(Network, TestStat, $\mathbf{z}, \boldsymbol{\rho}, \boldsymbol{\pi}$) - LogDataLikelihood(Network, TestStat, $\mathbf{z}', \boldsymbol{\rho}', \boldsymbol{\pi}'$)
    if (log(runif(1))< LogAcceptRate) **then**
      $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}'$
      $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi}'$
      $\mathbf{z} \leftarrow \mathbf{z}'$
      rhoTrace $\leftarrow$ cbind(rhoTrace, $\boldsymbol{\rho}$)
      piTrace $\leftarrow$ cbind(piTrace, $\mathbf{r}$)
    end if
  end for
  $\boldsymbol{\rho} \leftarrow$ rowMeans(rhoTrace[, nSave])
  $\boldsymbol{\pi} \leftarrow$ rowMeans(piTrace[, nSave])**return** $\boldsymbol{\pi}$
end function

Algorithm 5. Function: updating $\mathbf{z}|\tilde{\theta}$ by Swendsen-Wang

function SW(Network, $\mathbf{z}, \mathbf{r}, \widetilde{\theta}, \boldsymbol{\rho}, \boldsymbol{\pi}$)
  $G = <V, E> \leftarrow$ as.GraphObject(Network)
  procedure (graph clustering)
    $G \leftarrow G_{-1} \cup G_0 \cup G_1$; where $\forall$ node $i \in G_k = <V_k, E_k>, z_i = k$
    for $l \leftarrow \{-1, 0, 1\}$ **do**
      for all $e \in E_l$ **do**
        $W_e \leftarrow$ runif(1, 0, exp(ρ_{z_l}))
        if $(W_e < 1)$ **then** $e \leftarrow$ NULL
        end if
      end for
      $G_l \leftarrow \cup_{s=1}^{n_l} G_{ls}$, $G_{ls} = <V_{ls}, E_{ls}>$
    end for
    $G \leftarrow \cup_{l=-1}^{1} \cup_{s=1}^{n_l} G_{ls}$, $G_{ls} = <V_{ls}, E_{ls}>$
  end procedure
  procedure (graph relabling)
    for all $G_{cluster} = <V_{cluster}, E_{cluster}> \in \{G_{ls} = <V_{ls}, E_{ls}>, l = -1, 0, 1, s = 1, 2, \ldots, n_l\}$ **do**
      $\mathbf{z}'_{i \in G_{cluster}} \leftarrow$ SampleFromPosteriorDistributionOfZ($\mathbf{r}, \mathbf{z}, \widetilde{\theta}$)
    end for
    return z
  end procedure
end function

Algorithm 6. Procedure: update $\tilde{\theta}|\mathbf{z}$ via DPM fitting

function DPMDensityFitting(Network, $\mathbf{z}, \mathbf{r}$, PriorForDPMDensityFitting, ParaForMCMC)
  for z in $\{-1, 0, 1\}$ **do**
    Nodes $\leftarrow \{i\}_{\forall i, z_i = z}$
    DPMFit $\leftarrow$ DPDensityFitting($\{r_i\}_{i \in Nodes}$, PriorForDPMDensityFitting, ParaForMCMC)
    DPMFitSort $\leftarrow$ DPMFitClusterSortByMeanLocation(DPMFit)
    $\widetilde{\theta}_z \leftarrow$ DPMFitSort.Para
    $\{g_i\}_{\forall i, i \in \text{Nodes}} \leftarrow$ DPMFitSort.ClusterIndex
  end for
  return $\widetilde{\theta}, \mathbf{g}$
end function

Algorithm 7. Missing data imputation algorithm

function MissingDataImputation(MissingDataImputationMethod=c('FullyBayesianInference', 'NearestNeighborImpute'), Network,$\mathbf{r}, \mathbf{g}, \widetilde{\theta}$)
  if (MissingDataImputationMethod=='FullyBayesianInference') **then**
    for *loc* in $\{i\}_{\forall i, r_i \in \mathbf{r}_{mis}}$ **do**
      $r_{loc} \leftarrow \text{rnorm}(\theta_{g_{loc}})$
    end for
  end if
  if (MissingDataImputationMethod=='NearestNeighborImpute') **then**
    for *loc* in $\{i\}_{\forall i, r_i \in \mathbf{r}_{mis}}$ **do**
      Nbrs $\leftarrow$ ExtractNeighborsFromNetwork(Network)
      $r_{loc} \leftarrow \frac{1}{|\text{Nbrs}|} \sum_{k=1}^{|\text{Nbrs}|} r_k$
    end for
  end if
  return $\mathbf{r}_{mis}$
end function

The full conditional distribution for $\mathbf{z}$ given $\mathbf{W}$ is:

$$P(\mathbf{z}|\mathbf{W}, \mathbf{r}, \tilde{\theta}) \propto P(\mathbf{W}|\mathbf{z}) P(\mathbf{r}|\mathbf{z}, \tilde{\theta}) P(\mathbf{z}) \propto P(\mathbf{r}|\mathbf{z}, \tilde{\theta}) \exp\left[\sum_{i=1}^{n} (\tilde{\omega}_i \log(\pi_{z_i}))\right] \tag{A1}$$

DPM density updating Consider gene i with class indicator k and all the other genes with the same class indicator, if we integrate over $\mathbf{q}_k$, then the cluster index g_i has the following distribution:

$$\begin{aligned} P(g_i = g | g_1, g_2, \ldots, g_{i-1}) &= \frac{P(g_1, g_2, \ldots, g_{i-1}, g_i = g)}{P(g_1, g_2, \ldots, g_{i-1})} \\ &= \frac{\int_{(g_1, g_2, \ldots, g)} \Gamma(\tau_k) \Gamma(\tau_k / L_k)^{-L_k} g_1^{(\tau_k/L_k)-1} \cdots g_{L_k}^{(\tau_k/L_k)-1} dg_1 g_2 \cdots g_{L_k}}{\int_{(g_1, g_2, \ldots, g_{i-1})} \Gamma(\tau_k) \Gamma(\tau_k / L_k)^{-L_k} g_1^{(\tau_k/L_k)-1} \cdots g_{L_k}^{(\tau_k/L_k)-1} dg_1 g_2 \cdots g_{L_k}} \\ &= \frac{n_{i,g} + \tau_k / L_k}{i - 1 + \tau_k} \end{aligned}$$

where $n_{i,g} = \sum_{j=1}^{i-1} I[g_j = g]$ denotes the count of $g_j, j < i$ such that $g_j = g$.

Then let $L_k \to \infty$:

$$P(g_i = g, |g_1, g_2, \ldots, g_{i-1} \,\&\, g \in (g_1, \ldots, g_{i-1})) \to \frac{n_{i,g}}{i - 1 + \tau_k}$$
$$P(g_i = g, |g_1, g_2, \ldots, g_{i-1} \,\&\, g \notin (g_1, \ldots, g_{i-1})) \to \frac{\tau_k}{i - 1 + \tau_k} \tag{A2}$$