HHS Public Access

Author manuscript

Stat Anal Data Min. Author manuscript; available in PMC 2018 January 10.

Published in final edited form as:

Stat Anal Data Min. 2016 April ; 9(2): 106–116. doi:10.1002/sam.11306.

Nonlinear Joint Latent Variable Models and Integrative Tumor Subtype Discovery

Binghui Liu[1,2,3,*], **Xiaotong Shen**[2], and **Wei Pan**[3,*]

[1]School of Mathematics and Statistics, Northeast Normal University, Changchun, 130024 Jilin Province, China

[2]School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA

[3]Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, MN 55455, USA

Abstract

Integrative analysis has been used to identify clusters by integrating data of disparate types, such as deoxyribonucleic acid (DNA) copy number alterations and DNA methylation changes for discovering novel subtypes of tumors. Most existing integrative analysis methods are based on joint latent variable models, which are generally divided into two classes: joint factor analysis and joint mixture modeling, with continuous and discrete parameterizations of the latent variables respectively. Despite recent progresses, many issues remain. In particular, existing integration methods based on joint factor analysis may be inadequate to model multiple clusters due to the unimodality of the assumed Gaussian distribution, while those based on joint mixture modeling may not have the ability for dimension reduction and/or feature selection. In this paper, we employ a nonlinear joint latent variable model to allow for flexible modeling that can account for multiple clusters as well as conduct dimension reduction and feature selection. We propose a method, called integrative and regularized generative topographic mapping (irGTM), to perform simultaneous dimension reduction across multiple types of data while achieving feature selection separately for each data type. Simulations are performed to examine the operating characteristics of the methods, in which the proposed method compares favorably against the popular iCluster that is based on a linear joint latent variable model. Finally, a glioblastoma multiforme (GBM) dataset is examined.

Keywords

GTM; integrative clustering; latent variable models; tumor subtypes

1. INTRODUCTION

With the rapid development of microarray technologies, many molecular changes become possible to be monitored at the DNA and RNA levels. In addition to gene expression, genome-wide data, capturing both DNA methylation changes and DNA copy number

Correspondence to: Binghui Liu (liubh100@nenu.edu.cn), Wei Pan (weip@biostat.umn.edu).

alterations, are also available for the same biological samples [1–3]. As advocated in [4–6], integrative analysis incorporating multiple data types simultaneously offers a novel characterization of tumor etiology, often based on a joint probability model [7–10].

Commonly used methods for integrative analysis assume a joint probability Model, such as a latent variable model [8–11], representing the joint distribution of the multiple types of data over a common lower-dimensional space defined by latent variables. On this ground, integrative clustering is performed over the lower-dimensional space, where a latent vector can be considered a cluster indicator vector. Generally, there are two main approaches to formulating a latent variable model for integrative analysis: joint factor analysis [8,10] and joint mixture modeling [7]. Joint factor analysis introduces a continuous parameterization of the cluster indicator vector and further assumes that the continuous parameterization follows a Gaussian distribution with a zero mean and an identity covariance matrix [8,10]; in contrast, joint mixture modeling directly defines the cluster indicator vector to be a discrete latent vector [7]. Despite their successes, many issues remain. For joint factor analysis, one issue is that the conventional Gaussian latent variable model may be inadequate to model multiple clusters due to its unimodality. For mixture-based analysis, existing methods, for example, the integrative method in ref. [7], cannot perform feature selection and dimension reduction.

This article introduces a new framework of integrative modeling to account for multiple clusters and conduct dimension reduction and feature selection. This framework involves a nonlinear Gaussian mixture latent model with regularization, called integrative and regularized generative topographic mapping (irGTM). We specifically propose a regularized log-likelihood to integrate multiple types of data, achieving dimension reduction by seeking a common latent vector across all data types while simultaneously performing feature selection on each data type by regularizing the model parameters. As showed by our numerical results, irGTM yields a great improvement over existing methods in terms of clustering accuracy.

The major contributions of irGTM are two-fold: modeling multiple integrative clusters (i.e. clusters defined by multiple types of data) while achieving feature selection, which tends to overcome the difficulty that existing integrative analysis methods are incapable of achieving the two goals simultaneously. Although the nonlinear framework of GTM [12] may describe and distinguish clusters with special shapes, it is greatly inefficient in computation due to overparametrization. Therefore, to be computationally feasible in high-dimensional situations, we modify GTM to model multiple clusters using a smaller number of parameters Based on this, some examples will be studied subsequently to investigate the performance of irGTM with respect to distinguishing multiple integrative clusters or subtypes of cancers.

Note that in addition to clustering analysis as focused here, integrative analysis of multiple types of genomic data has also been developed and applied for other purposes, [13–16]. As these purposes are not our main concern, we will not introduce them in detail in this article.

This article is organized as follows: Section 2 introduces irGTM in detail. Section 3 compares irGTM with iCluster based on some benchmark examples, demonstrating the

advantage of the proposed method. In Section 4, we apply irGTM to integrate gene expression, DNA methylation and copy number data for subtype discovery based on a glioblastoma multiforme dataset.

2. METHODS

2.1. Joint probability model

Given multiple types of data expressed as $\mathscr{X} = \{\mathscr{X}^{(s)}, s \in \{1, \ldots, S\}\}$, where each $s \in \{1, \ldots, S\}$ indexes one type of data, and $\mathscr{X}^{(s)} = \{\boldsymbol{X}_1^{(s)}, \cdots, \boldsymbol{X}_N^{(s)}\}$ is observed data of type s, with $\boldsymbol{X}_n^{(s)} = (X_{n1}^{(s)}, \cdots, X_{nD^{(s)}}^{(s)})^T$ for each $n \in \{1, \ldots, N\}$. Specifically, $\{\boldsymbol{X}_n^{(s)}, n \in \{1, \cdots, N\}\}$ are independent and identically distributed (i.i.d.) with the same distribution as a length-$D^{(s)}$ random vector $\boldsymbol{X}^{(s)} = (X_1^{(s)}, \cdots, X_{D^{(s)}}^{(s)})^T$.

In our framework, random vectors $\{\boldsymbol{X}^{(s)}, s \in \{1, \cdots, S\}\}$ are linked to a common latent vector $\boldsymbol{Z} = (Z_1, \cdots, Z_D)^T$, which has a lower-dimension with D possibly much smaller than $D^{(s)}$ for each $s \in \{1, \cdots, S\}$ and is introduced to simultaneously express the distributions of all data types, Based on this, we assume that the latent vector corresponds to the integrative clustering of the pooled multiple types of data $\mathscr{X}$.

We now introduce the framework of a joint probability model of these multiple types of data. First, assume that $\{\boldsymbol{X}^{(s)}, s \in \{1, \cdots, S\}\}$ are conditionally independent given the latent variables $\boldsymbol{Z}$, that is:

$$p\left(\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(S)} | \boldsymbol{Z}\right) = \prod_{s=1}^{S} p\left(\boldsymbol{X}^{(s)} | \boldsymbol{Z}\right), \quad (1)$$

where each conditional distribution $p(\boldsymbol{X}^{(s)}|\boldsymbol{Z})$ with $s \in \{1, \cdots, S\}$ is given through a nonlinear mapping $\boldsymbol{W}^{(s)}\boldsymbol{\phi}(\boldsymbol{Z})$ from the latent variables $\boldsymbol{Z}$ to the observed data variables $\boldsymbol{X}^{(s)}$:

$$\boldsymbol{X}^{(s)} = \boldsymbol{W}^{(s)}\phi(\boldsymbol{Z}) + \boldsymbol{E}^{(s)}. \quad (2)$$

In particular, for each $s \in \{1, \cdots, S\}$, $\boldsymbol{W}^{(s)}$ is a $D^{(s)} \times K$ coefficient matrix; $\boldsymbol{\phi}(\boldsymbol{Z}) = (\boldsymbol{\phi}_1(\boldsymbol{Z}), \cdots, \boldsymbol{\phi}_K(\boldsymbol{Z}))^T$ is a radially symmetric Gaussian basis function with

$$\phi_k(\boldsymbol{Z}) = \exp\left(-\frac{\|\boldsymbol{Z} - \boldsymbol{\mu}_k\|_2^2}{2\delta^2}\right) \quad (3)$$

for each $k \in \{1, \cdots, K\}$, and $\{\boldsymbol{E}^{(s)}, s \in \{1, \cdots, S\}\}$ are $\boldsymbol{Z}$-independent noise random vectors following mutually independent isotropic Gaussian distributions with variances $\{\sigma^{(s)2}, s \in \{1, \cdots, S\}\}$, respectively, where K denotes the number of integrative clusters of the multiple

types of data $\mathcal{X}$, $\delta > 0$ is the scale of the Gaussian basis function, and $\{\boldsymbol{\mu}_k, k \in \{1, \cdots, K\}\}$ are the centers corresponding to the K integrative clusters. As the dependent variable of the proposed Gaussian basis functions is the latent vector $\boldsymbol{Z}$, where all the possible values of $\boldsymbol{Z}$ belong to a sample-independent latent space, we set the radius of the latent space as the scale, which seems to perform well as shown in the simulation results in Section 3.

Generally, in a latent variable model of (integrative) clustering analysis, such as iCluster [8], a length-(K−1) continuous parameterization Z^* following a Gaussian distribution is used to replace the cluster indicator vector $(\boldsymbol{I}[\{\boldsymbol{X}^{(s)}, s \in \{1, \cdots, S\}\}$ belongs to cluster 1], ⋯, $\boldsymbol{I}[\{\boldsymbol{X}^{(s)}, s \in \{1, \cdots, S\}\}$ belongs to cluster $K]\big)^T$ for computational reasons, and then, clustering is achieved based on the posterior mean of Z^*. As a Gaussian distribution implies only one mode or center, a continuous parameterization of Z^* by a Gaussian distribution may not be sufficient to represent multiple clusters. As in a K-means algorithm or a Gaussian mixture model, multiple centers are introduced to describe multiple clusters, suggesting that a more flexible parameterization with multiple centers may perform better than that with a single center. Based on this, we propose using the radially symmetrical Gaussian basis functions $\boldsymbol{\phi}(\boldsymbol{Z})$ with K centers $\{\boldsymbol{\mu}_k, k \in \{1, \cdots, K\}\}$ to approximate the cluster indicator vector.

Next, we assume that $\boldsymbol{Z}$ follows a discrete uniform distribution:

$$p(\boldsymbol{Z}) = \frac{1}{M} \sum_{m=1}^{M} \delta(\boldsymbol{Z} - \boldsymbol{\nu}_m), \tag{4}$$

where for each $m \in \{1, \cdots, M\}$, $\delta(\boldsymbol{Z} - \boldsymbol{\nu}_m) = 1$ if $\boldsymbol{Z} = \boldsymbol{\nu}_m$, otherwise $\delta(\boldsymbol{Z} - \boldsymbol{\nu}_m) = 0$. We let the latent space be a unit circle on $\mathbb{R}^2$, and let the latent-space sample points $\{\boldsymbol{\nu}_m, m \in \{1, \cdots, M\}\}$ be spread uniformly along the circle, that is, $\boldsymbol{\nu}_m = (\cos(2\pi(m-1)/M), \sin(2\pi(m-1)/M))^T$ for each $m \in \{1, \cdots, M\}$. Accordingly, we let $\boldsymbol{\mu}_k = (\cos(2\pi(k-1)/K), \sin(2\pi(k-1)/K))^T$ for each $k \in \{1, \cdots, K\}$, which are the most mutually exclusive K points on the unit circle. As suggested by ref. [12], the choice of the number M and the locations of the latent-space sample points are not critical. Therefore, hereafter, we set the latent-space sample points and the centers of the basis functions as above and fix $M = 100$

Combining (1), (2) and (4) leads to the joint probability distribution of $\{\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(S)}\}$ as follows:

$$p\left(\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(S)}; \left\{\boldsymbol{W}^{(s)}\right\}_{s=1}^{S}, \left\{\sigma^{(s)2}\right\}_{s=1}^{S}\right) = \frac{1}{M} \sum_{m=1}^{M} \prod_{s=1}^{S} p\left(\boldsymbol{X}^{(s)} | \boldsymbol{\nu}_m; \boldsymbol{W}^{(s)}, \sigma^{(s)2}\right), \tag{5}$$

where

$$p\left(\boldsymbol{X}^{(s)} | \boldsymbol{\nu}_m; \boldsymbol{W}^{(s)}, \sigma^{(s)2}\right) = p\left(\boldsymbol{X}^{(s)} | \boldsymbol{Z} = \boldsymbol{\nu}_m; \boldsymbol{W}^{(s)}, \sigma^{(s)2}\right)$$

is a Gaussian distribution with a mean vector of $\boldsymbol{W}^{(s)}\boldsymbol{\phi}(\boldsymbol{\nu}_m)$ and covariance matrix of $\sigma^{(s)2}I_{D^{(s)}}$, and $I_{D^{(s)}}$ is an identity matrix of size $D^{(s)}$. Given the observed multiple types of data $\mathcal{X} = \{\mathcal{X}^{(s)}, s \in \{1, \ldots, S\}\}$, we estimate $\{\boldsymbol{W}^{(s)}\}_{s=1}^{S}$ and $\{\sigma^{(s)2}\}_{s=1}^{S}$ by maximizing the log-likelihood:

$$\mathcal{L}\left(\left\{\boldsymbol{W}^{(s)}\right\}_{s=1}^{S}, \left\{\sigma^{(s)2}\right\}_{s=1}^{S}\right) = \sum_{m=1}^{N} \ln \left\{\frac{1}{M}\sum_{m=1}^{M}\prod_{s=1}^{S} p\left(\boldsymbol{X}_n^{(s)} | \boldsymbol{\nu}_m; \boldsymbol{W}^{(s)}, \sigma^{(s)2}\right)\right\}. \quad (6)$$

Maximization may proceed by using the expectation maximization (EM) algorithm [17] to deal with the latent variables, which we will elaborate next.

2.2. The EM algorithm

To detail our EM algorithm, we first introduce the E-step. Let $\{\boldsymbol{Z}_n, n \in \{1, \cdots, N\}\}$ denote the latent vectors of all the samples in $\mathcal{X}$, which are independent and identically distributed (i. i. d.) with the same distribution as $\boldsymbol{Z}$. Then, the complete-data log-likelihood is

$$\mathcal{L}_c(\{\boldsymbol{W}^{(s)}\}_{s=1}^{S}, \{\sigma^{(s)2}\}_{s=1}^{S}) = \sum_{n=1}^{N}\sum_{s=1}^{S} \ln \left\{p\left(\boldsymbol{X}_n^{(s)} | \boldsymbol{Z}_n; \{\boldsymbol{W}^{(s)}\}_{s=1}^{S}, \{\sigma^{(s)2}\}_{s=1}^{S}\right) \times \frac{1}{M}\sum_{m=1}^{M}\delta(\boldsymbol{Z}_n - \boldsymbol{\nu}_m)\right\}. \quad (7)$$

To seek sparse estimates of the coefficient matrices $\{\boldsymbol{W}^{(s)}\}_{s=1}^{S}$, we employ an L_1-penalized complete-data log-likelihood as in ref. [8]:

$$\mathcal{L}_{c,p}\left(\left\{\boldsymbol{W}^{(s)}\right\}_{s=1}^{S}, \left\{\sigma^{(s)2}\right\}_{s=1}^{S}\right) \triangleq \mathcal{L}_c\left(\left\{\boldsymbol{W}^{(s)}\right\}_{s=1}^{S}, \left\{\sigma^{(s)2}\right\}_{s=1}^{S}\right) - \sum_{s=1}^{S} J_{\lambda(s)}\left(\boldsymbol{W}^{(s)}, \sigma^{(s)2}\right), \quad (8)$$

where

$$J_{\lambda_s}\left(\boldsymbol{W}^{(s)}, \sigma^{(s)2}\right) = \frac{\lambda^{(s)}}{\sigma^{(s)2}}\sum_{d=1}^{D^{(s)}}\sum_{k=1}^{K}|W_{dk}^{(s)}| \quad (9)$$

is the L_1-penalty of the sth data type with a non-negative tuning parameter $\lambda^{(s)}$, controlling its model complexity ref. [18].

Given the current coefficient matrices $\{\boldsymbol{W}_{old}^{(s)}\}_{s=1}^{S}$ and the current noise variances $\{\sigma_{old}^{(s)2}\}_{s=1}^{S}$, the conditional expectation of the penalized complete-data log-likelihood is as follows:

$$\sum_{s=1}^{S}\sum_{n=1}^{N}\sum_{m=1}^{M} R_{mn}\left(\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right)\ln\left\{p(\boldsymbol{X}_n^{(s)}|\boldsymbol{\nu}_m;\boldsymbol{W}^{(s)},\sigma^{(s)2})\right\}-\sum_{s=1}^{S}\frac{\lambda^{(s)}}{\sigma^{(s)2}}\sum_{d=1}^{D^{(s)}}\sum_{k=1}^{K}|W_{dk}^{(s)}|, \tag{10}$$

where for each $m \in \{1, \cdots, M\}$ and each $n \in \{1, \cdots, N\}$, the posterior probability

$$R_{mn}\left(\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right) \triangleq p\left(\boldsymbol{\nu}_m|\left\{\boldsymbol{X}_n^{(s)}\right\}_{s=1}^{S};\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right)$$
$$=\frac{\prod_{s=1}^{S}p\left(\boldsymbol{X}_n^{(s)}|\boldsymbol{\nu}_m;\boldsymbol{W}_{old}^{(s)},\sigma_{old}^{(s)2}\right)}{\sum_{m'=1}^{M}\prod_{s=1}^{S}p\left(\boldsymbol{X}_n^{(s)}|\boldsymbol{\nu}_{m'};\boldsymbol{W}_{old}^{(s)},\sigma_{old}^{(s)2}\right)} \tag{11}$$

is evaluated using Bayes' theorem.

For the M-step, note that maximizing (10) with respect to $\{\boldsymbol{W}^{(s)}, s \in \{1, \cdots, S\}\}$ is equivalent to minimizing S L_1-penalized least squares problems separately. As a result, for each $s \in \{1, \cdots, S\}$, the solution $\boldsymbol{W}_{new}^{(s)}$ can be evaluated using some existing software for a penalized least squares problem, such as glmnet package of R or Matlab. Alternatively, we can simply use the soft-thresholding estimator $\boldsymbol{W}_{newST}^{(s)}$ as an approximation:

$$\boldsymbol{W}_{newST}^{(s)}=\text{sign}\left(\boldsymbol{W}_{new}^{*(s)}\right)\left(|\boldsymbol{W}_{new}^{*(s)}|-\lambda^{(s)}\right)_{+}, \tag{12}$$

with

$$\boldsymbol{W}_{new}^{*(s)}=\left(\left(\boldsymbol{\Phi}^T\boldsymbol{G}_{old}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{R}_{old}\boldsymbol{X}^{(s)}\right)^T, \tag{13}$$

where sign(·)(|·| − $\lambda^{(s)}$)$_+$ is applied component-wise; $\boldsymbol{\Phi}$ is an $M \times K$ matrix with elements $\Phi_{mk} = \phi_k(\boldsymbol{\nu}_m)$, $m \in \{1, \cdots, M\}$, $k \in \{1, \cdots, K\}$; $\boldsymbol{X}^{(s)}$ is an $N \times D^{(s)}$ matrix with elements $X_{nd}^{(s)}$, $n \in \{1, \cdots, N\}$, $d \in \{1, \cdots, D^{(s)}\}$; $\boldsymbol{R}_{old}$ is an $M \times N$ matrix with elements $R_{mn}\left(\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right)$, $m \in \{1, \cdots, M\}$, $n \in \{1, \cdots, N\}$; and $\boldsymbol{G}_{old}$ is an $M \times M$ diagonal matrix with elements

$$(G_{old})_{mm}=\sum_{n=1}^{N}R_{mn}\left(\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right), \quad (14)$$

$m \in \{1, \cdots, M\}$. In particular, (13) can be solved using standard matrix inversion methods based on the singular value decomposition to allow for possible ill conditioning. Moreover, maximizing (10) in $\{\sigma^{(s)2}, s \in \{1, \cdots, S\}\}$ leads to the following updating formula based on $\{\boldsymbol{W}_{new}^{(s)}\}_{s=1}^{S}$:

$$\sigma_{new}^{(s)2}=\frac{1}{ND^{(s)}}\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}R_{mn}\left(\left\{\boldsymbol{W}_{old}^{(s)}\right\}_{s=1}^{S},\left\{\sigma_{old}^{(s)2}\right\}_{s=1}^{S}\right)\times\|\boldsymbol{W}_{new}^{(s)}\phi(\boldsymbol{\nu}_m)-\boldsymbol{X}_n^{(s)}\|_2^2+2\lambda^{(s)}\sum_{d=1}^{D^{(s)}}\sum_{m=1}^{M}\left|(W_{new}^{(s)})_{dm}\right|\right\}, \quad (15)$$

for each $s \in \{1, \cdots, S\}$. The EM algorithm iterates between the E-step and M-step until convergence, and we use $\left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^{S}$ and $\left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^{S}$ to denote the converged estimates.

In the above EM algorithm, we initialize $\boldsymbol{W}^{(1)}$ so that the GTM model of $\boldsymbol{X}^{(1)}$ initially approximates the corresponding principal component analysis (PCA) following the initialization method in ref. [12]. To be specific, we first evaluate the sample covariance matrix of $\mathscr{X}^{(1)}$, compute its first and second principal eigenvectors, and then determine the initial estimate of $\boldsymbol{W}^{(1)}$ by minimizing the following error function:

$$Err^{(1)}=\frac{1}{2}\sum_{m=1}^{M}\|W^{(1)}\phi(\boldsymbol{\nu}_m)-\boldsymbol{U}^{(1)}\boldsymbol{\nu}_m\|_2^2, \quad (16)$$

where the columns of $\boldsymbol{U}^{(1)}$ are given by the two eigenvectors. As suggested by ref. [12], this represents the sum-of-squares error between the projections of the latent points into the data space by the GTM model of $\boldsymbol{X}^{(1)}$ and the corresponding projections obtained from PCA. Next, we initialize each $\boldsymbol{W}^{(s)}$ with $s \in \{2, \cdots, S\}$ as a $D^{(s)} \times K$ matrix with zero-elements. In addition, for each $s \in \{1, \cdots, S\}$, we initialize $\sigma^{(s)2}$ as the third eigenvalue of the sample covariance matrix of $\mathscr{X}^{(s)}$, representing the variance of $\mathscr{X}^{(s)}$ away from the corresponding PCA plane of $\mathscr{X}^{(s)}$. It is worth noting that based on our limited experience, initializing each $\boldsymbol{W}^{(s)}$ with $s \in \{2, \cdots, S\}$ using the same method as in (16), irGTM often performs poorly. This is possibly because separate estimates of $\boldsymbol{W}^{(s)}$ may not contribute to a common latent vector $\boldsymbol{Z}$ across all the data types. As suggested by the following simulation studies, clustering performance of irGTM is quite similar to that based on different orders of data types.

2.3. Tuning parameters

Here, we introduce a resampling-based procedure for selecting the regularization parameters, similar to that of [19]. The procedure partitions the multiple types of data into a training set and a test set iteratively. In each iteration, we first train the irGTM model using the training set and let $\left\{\hat{\boldsymbol{W}}_{tr}^{(s)}\right\}_{s=1}^{S}$ and $\left\{\hat{\sigma}_{tr}^{(s)2}\right\}_{s=1}^{S}$ be the corresponding estimators. Then, for each observation $\left\{\boldsymbol{X}_{n_{te}}^{(1)},\cdots,\boldsymbol{X}_{n_{te}}^{(S)}\right\}$ in the test set, we compute the posterior mean of the latent variables

$$\left\langle \boldsymbol{Z}|\left\{\boldsymbol{X}_{n_{te}}^{(s)}\right\}_{s=1}^{S};\left\{\hat{\boldsymbol{W}}_{tr}^{(s)}\right\}_{s=1}^{S},\left\{\hat{\sigma}_{tr}^{(s)2}\right\}_{s=1}^{S}\right\rangle = \int p\left(\boldsymbol{z}|\left\{\boldsymbol{X}_{n_{te}}^{(s)}\right\}_{s=1}^{S};\left\{\hat{\boldsymbol{W}}_{tr}^{(s)}\right\}_{s=1}^{S},\left\{\hat{\sigma}_{tr}^{(s)2}\right\}_{s=1}^{S}\right)\boldsymbol{z}d\boldsymbol{z}$$
$$=\sum_{m=1}^{M} R_{mn_{te}}\left(\left\{\hat{\boldsymbol{W}}_{tr}^{(s)}\right\}_{s=1}^{S},\left\{\hat{\sigma}_{tr}^{(s)2}\right\}_{s=1}^{S}\right)\boldsymbol{\nu}_m. \quad (17)$$

Next, we compute the Euclidean distance $d_{te|tr}$ of the posterior means. We denote by $ne_{te|tr}^{N_{ne}}(n_{te})$ the N_{ne}-nearest neighbors of the posterior mean of the latent variables for each n_{te} in the index set of the test set, where N_{ne} is a given number smaller than the sample size of the test set. On the other hand, we train the irGTM model using the test set, compute the corresponding posterior mean of the latent variables for each observation indexed by n_{te} in the test set, and then compute the corresponding Euclidean distance d_{te} of the posterior means. Let $ne_{te}^{N_{ne}}(n_{te})$ denote the N_{ne}-nearest neighbors of the posterior mean of the latent variables for each n_{te} in the index set of the test set.

Then, we define the prediction strength as follows:

$$\frac{1}{N_{te}}\sum_{n_{te}}\frac{\left|\mathrm{ne}_{te|tr}^{N_{ne}}(n_{te})\cap \mathrm{ne}_{te}^{N_{ne}}(n_{te})\right|}{N_{ne}}, \quad (18)$$

where $|\cdot|$ represents the number of elements in a set. For selection of the penalty parameters $\{\lambda^{(s)}, s\in S\}$, we choose the ones with the highest average prediction strength.

Note that while the penalty parameters are large enough, the posterior means of the latent variables of the points in the test set will converge to one point. Then, $\mathrm{ne}_{te|tr}^{N_{ne}}(n_{te})$ and $\mathrm{ne}_{te}^{N_{ne}}(n_{te})$ will be the same if they are obtained by choosing N_{ne} points with the lowest indices in the test set without n_{te}, which forces the prediction strength to be 1. To avoid such a situation, we add small random Gaussian noise to the posterior mean of the latent variables of each data point in the test set before computing the Euclidean distance $d_{te|tr}$ and d_{te}.

This procedure is used to select the regularization parameters in a situation where the number of clusters K is fixed in advance. When K is unknown, we may use a method that is commonly used in determining the number of clusters in clustering analysis, such as the highest Silhouette index [20]. To be specific, for each $K > 1$ in $\mathcal{K}$ (a candidate set of positive numbers), we obtain the resulting clustering assignment $I(K)$ by fixing the number of clusters as K and then select $\hat{K} \in \mathcal{K}$ with the highest Silhouette index as the optimal estimate of the number of clusters, based on which, we obtain the final estimate of the clustering assignment, say $I(\hat{K})$.

2.4. Integrative clustering

The goal of irGTM is for integrative clustering, which is achieved by using Bayes' theorem, to invert the transformation from the latent space to the data space. To be specific, for each data point $\{\boldsymbol{X}_n^{(s)}\}_{s=1}^S$ with $n \in \{1, \cdots, N\}$, we summarize the posterior distribution by the posterior mean:

$$\left\langle \boldsymbol{Z} | \left\{\boldsymbol{X}_n^{(s)}\right\}_{s=1}^S; \left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^S, \left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^S \right\rangle = \sum_{m=1}^M R_{mn} \left(\left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^S, \left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^S \right) \boldsymbol{\nu}_m. \quad (19)$$

These posterior means are used for dimension reduction and then integrative clustering. The cluster memberships will then be specifically figured out by applying a standard K-means clustering algorithm [21] on the posterior means.

Alternatively, we can compute the posterior mean of $\boldsymbol{\phi}(\boldsymbol{Z})$ rather than of $\boldsymbol{Z}$:

$$\left\langle \phi(\boldsymbol{Z}) | \left\{\boldsymbol{X}_n^{(s)}\right\}_{s=1}^S; \left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^S, \left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^S \right\rangle = \sum_{m=1}^M R_{mn} \left(\left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^S, \left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^S \right) \boldsymbol{\phi}(v_m),$$

(20)

and assign $\{\boldsymbol{X}_n^{(s)}\}_{s=1}^S$ to cluster $k \in \{1, \cdots, K\}$ whenever the kth element of

$$\left\langle \phi(\boldsymbol{Z}) | \left\{\boldsymbol{X}_n^{(s)}\right\}_{s=1}^S; \left\{\hat{\boldsymbol{W}}^{(s)}\right\}_{s=1}^S, \left\{\hat{\sigma}^{(s)2}\right\}_{s=1}^S \right\rangle$$

is the largest element of the vector as $\boldsymbol{\phi}(\boldsymbol{Z})$ is an approximation of the cluster indicator vector.

3. SIMULATION STUDIES

This section performs simulations to examine the operating characteristics of the proposed method and compare it against its competitors: iCluster [8], naive integration of PCA (NI)

[22], and several methods applicable to two types of data, such as partial least squares regression (PLS) [23], co-inertia analysis (CIA) [24], and canonical correlation analysis (CCA) [25] in terms of clustering accuracy measured by the Rand index and the adjusted Rand index [26]. Note that we exclude comparison with ref. [7] as the latter is not designed for cases involving many non-informative variables.

3.1. Simulation set-ups

Three examples are considered, which are based on benchmark examples in refs. [8,10]. In these examples, for each type of data, the cluster information is associated with a small number of variables that are thought of as clustering features, based on which one cluster is distinguishable from the other two that remain non-separable.

Case 1: A random sample of $n = 150$ is taken from three clusters with $\{1, \cdots, 50\}$, $\{51, \cdots, 100\}$, and $\{101, \cdots, 150\}$ from clusters 1–3, respectively. Let $D^{(1)} = D^{(2)} = 500$ and $\mu = 1.5$. For $s = 1$, $X_{ij}^{(1)} \sim \mathcal{N}(\mu, 1)$; $i = 1, \cdots, 50$, $j = 1, \cdots, 10$; $X_{ij}^{(1)} \sim \mathcal{N}(1, 1)$; $i = 51, \cdots, 100$, $j = 101, \cdots, 110$; and $X_{ij}^{(1)} \sim \mathcal{N}(0, 1)$ for the rest. For $s = 2$, $X_{ij}^{(2)} = 0.5X_{ij}^{(1)} + \varepsilon_{ij}$, where $\varepsilon_{ij} \sim \mathcal{N}(0, 1)$; $i = 1, \cdots, 50$, $j = 1, \cdots, 10$; $X_{ij}^{(2)} \sim \mathcal{N}(\mu, 1)$; $i = 101, \cdots, 150$, $j = 101, \cdots, 110$; and $X_{ij}^{(2)} \sim \mathcal{N}(0, 1)$ for the rest. Interestingly, the two types of data in Case 1 are correlated in the first 10 dimensions. In addition, for each data type, there are two groups of features, $\{1, \cdots, 10\}$ and $\{101, \cdots, 110\}$, which allow us to discriminate one cluster from the other two.

Cases 2 and 3 are similar to Case 1, except $\mu = 1.3$ and $\mu = 1.1$, respectively. As the value of μ determines the strength of clustering features, these cases are used to investigate the clustering performance of the methods in the presence of weaker clustering features.

To guard against potential confounding due to different choices of the number of integrative clusters K, we fix K at 3 for Cases 1–3. To choose the tuning parameter $\lambda = (\lambda^{(1)}, \cdots, \lambda^{(S)})^T \in \Lambda \subseteq \mathbb{R}^S$ for irGTM, we use the resampling-based procedure in Section 2.3 with $|\Lambda| = |\{\lambda^{(1,1)}, \cdots, \lambda^{(1,8)}\} \times \cdots \times \{\lambda^{(S,1)}, \cdots, \lambda^{(S,8)}\}| = 8^S$ and for iCluster with $|\Lambda| = |\{\lambda_1, \cdots, \lambda_8\}| = 8$, which is set in default in the `R` package '`iCluster`'. Here, $|\cdot|$ denotes the number of elements in a set. All the simulations are performed on a PC with a single processor Intel(R) Core(TM) i7 CPU @ 3.40GHz (16G Memory), except that iCluster was run with four processors for parallel computation. For a fair comparison, reported run times for iCluster were simply four times the original run times.

Note that to investigate clustering performances of irGTM with different orders of data types, let irGTM_1 denote the proposed method of the order of $(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)})$ and irGTM_2 denote that of the order of $(\boldsymbol{X}^{(2)}, \boldsymbol{X}^{(1)})$.

3.2. Simulation results

Table 1 summarizes the results of all the methods considered in Cases 1–3 based on 100 repetitions respectively.

For Case 1, as indicated by Table 1, irGTM and iCluster outperform their competitors in terms of the accuracy of clustering, measured by the Rand index and adjusted Rand index. As suggested by panel (A) of Figure 1, irGTM and iCluster make the three clusters almost separable. Moreover, in view of panel (A) of Figure 2, we note that the informative variables $\{1, \cdots, 10\}$ and $\{101, \cdots, 110\}$ are almost correctly identified by irGTM. Note that this example was ideal for iCluster, which was used in [8] to demonstrate the advantages of iCluster over integration by PCA for each type of data. For Cases 2–3, iCluster and naive fail to capture the clustering structure due to weaker clustering information. Again, as demonstrated in Table 1, irGTM outperforms its competitors in terms of the clustering accuracy.

In summary, irGTM is competitive for high-throughput genomic data and is much more accurate than iCluster in integrative clustering of different types of data, especially in situations where clustering features are relatively weak. Moreover, it is more computationally more efficient than iCluster.

4. APPLICATIONS

We now consider a glioblastoma multiform (GBM) dataset generated by The Cancer Genome Atlas (TCGA) [27], which is available in the R package ‘iCluster’. Based on only 1,740 most variable expression levels [27], four distinct GBM subtypes, Proneural (P), Neural (N), Classical (C), and Mesenchymal (M), were identified.

The Kaplan–Meier (K–M) curves [28] are often used to confirm whether some patient groups identified by a clustering analysis have distinct survival outcomes. From the K–M plot of the four GBM subtypes (panel (e) of Figure 4), the survival probability of the Proneural subtype is higher than those of the other three subtypes, while the differences in survival among the latter three subtypes are small. For instance, after 2 years, about 50% patients in the Proneural subgroup survive, but only less than 20% patients in each of the latter three subgroups of GBM survive.

For the four subtypes, the overall differences among the four survival curves are only Moderate, with a relatively high p-value 0.018 from the Log-rank test, testing survival differences among the four patient groups [29].

Integrative analysis of gene expression data and other data may help identify novel subtypes with more distinguished survival outcomes. Shen *et al.* [8] performed an integrative analysis by iCluster, combining the use of DNA copy number, methylation, and mRNA expression in the GBM data. They divided 55 GBM patients into three integrative clusters (iClusters 1, 2, and 3). From panel (d) of Figure 4 (regenerated Figure 6 of ref. [9]), iCluster 1 is associated with a higher survival curve, while both iClusters 2 and 3 are associated with two lower and similar survival curves. Furthermore, the p-value of the Log-rank test is 0.003, much more significant than that of the above four subtypes.

To test the performance of the proposed method and to compare with iCluster, we analyze the same dataset with the same 55 GBM patients and three types of data. Details of the corresponding results are summarized in Figures 3 and 4, including the heat map of each

data type, plot of posterior means of the latent vector, and the survival curves (K–M plots) of the subtypes via iCluster and irGTM. From panel (c) of Figure 4, it seems that cluster 1 is associated with the highest survival curve, which mainly includes the Proneural (P) subtype; cluster 3 is associated with the lowest survival, and the survival of cluster 2 is intermediate between those of the other two clusters. In contrast to the three clusters identified by iCluster [9], the three clusters uncovered by the proposed method can be better distinguished from each other with more significant survival differences with a smaller p-value, 0.001, by the Log-rank test. Note that in this application, the results of irGTM with different orders of the three data types are almost the same and thus omitted.

5. DISCUSSIONS

5.1. Alternative choice of $\phi(Z)$

In this subsection, we investigate the performance of irGTM with an alternative choice of $\boldsymbol{\phi}(\boldsymbol{Z}) = (\phi_1(\boldsymbol{Z}), \cdots, \phi_K(\boldsymbol{Z}))^T$ in (2). In particular, for each $k \in \{1, \cdots, K\}$, $\phi_k(\boldsymbol{Z})$ is replaced by the angle distance between $\boldsymbol{Z}$ and μ_k,

$$\phi_k(\boldsymbol{Z}) = \arccos\left(\frac{\boldsymbol{Z}^T \mu_k}{\|\boldsymbol{Z}\| \|\mu_k\|}\right).$$

Now, reconsider all the three set-ups in Section 3 with the above angle distance to obtain quite similar Rand and adjusted Rand index results, measuring the integrative clustering performance, which is summarized in Table 2.

5.2. Alternative choice of latent space and centers of clusters

As mentioned in [12], in general, the performance of GTM does not greatly depend on the choice of latent space and its centers. Here, we investigate the performance of irGTM with an alternative choice of latent space and cluster centers. Here, we let the latent space be a three-dimensional unit sphere and choose $M = 100$ points uniformly from the surface of the unit sphere as the sample points. Then, we choose K mutually exclusive points on the surface of the unit sphere as the corresponding centers of the K clusters. We reconsider all the three set-ups in Section 3 with the above choice of latent space and cluster centers. From Table 3, the Rand and adjusted Rand index results are similar to those of the proposed method in Table 1. However, we see that the computational cost increases as the number of latent variables (or the dimension of the latent space) increases.

On the other hand, once the latent space is still a unit circle, and the latent-space sample points as well as its cluster centers perform a rotation on the circle, the clustering performance of the proposed method may be almost unchanged, which is supported by some additional numerical results not exhibited in this article to avoid reduplicate statements.

5.3. Alternative choice of basis functions

In this subsection, we rebuilt irGTM by replacing the basis functions of latent variables with the latent vector itself, that is, the cluster indicator vector. In addition, we let the latent space

be composed of all the values of the length-K cluster indicator, that is, $\{(1, 0, \cdots, 0)^T, (0, 1, 0, \cdots, 0)^T, \cdots, (0, \cdots, 0, 1)^T\}$, and just let the sample points be these values of the cluster indicator. We summarize the corresponding results in Table 4, where the computation of the rebuilt irGTM is improved, but the clustering performance becomes a bit worse.

5.4. Alternative choices of the scale of the basis function

In this subsection, we investigate the performance of irGTM with other choices of the scale of the basis function. From Table 5, in Cases 1–3, the performance of irGTM with the scale δ belonging to [0.5,1] is similar to that with $\delta = 1$.

Finally, based on all the exhibited numerical results in this section, in general, the clustering performance of the proposed method may not greatly depend on the choice of basis function, latent space and space centers, which implies that the proposed method may mainly benefit from using a discrete distribution of the latent variables, while a nonlinear framework with alternative choices of basis functions, latent space, and space centers may offer a more flexible and general model.

6. CONCLUSION

This paper introduces a novel integrative analysis method based on a nonlinear latent model, called irGTM, which is computationally efficient using an EM algorithm. It has one key feature in defining a novel parameterization to better approximate the latent cluster indicators by introducing a discrete latent space as well as a nonlinear mapping from the latent space to each data space. This permits flexible modeling to account for multiple clusters and to perform dimension reduction and feature selection. As a result, it may improve the performance for integrative clustering.

Acknowledgments

We thank the editor and the reviewers for helpful comments and suggestions. This work was supported by NIH grants R01-GM081535, R01-GM113250 and R01-HL105397, by NSF grants DMS-0906616 and DMS-1207771 and by NSFC grant 11571068.

References

1. Holm K, Hegardt C, Staaf J, et al. Molecular subtypes of breast cancer are associated with characteristic DNA methylation patterns. Breast Cancer Res. 2010; 12:R36. [PubMed: 20565864]
2. Jones P, Baylin S. The fundamental role of epigenetic events in cancer. Nat Rev Genet. 2002; 3:415–428. [PubMed: 12042769]
3. Pollack JR, Słrlie T, Perou CM, et al. Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. Proc Natl Acad Sci. 2002; 99:12963–12968. [PubMed: 12297621]
4. Maria EF, Mark R, Reid FT, et al. An integrative genomic and epigenomic approach for the study of transcriptional regulation. PLoS One. 2008; 3:e1882. [PubMed: 18365023]
5. Menezes RX, Boetzer M, Sieswerda M, et al. Integrated analysis of DNA copy number and gene expression microarray data using gene sets. BMC Bioinformatics. 2009; 10:203. [PubMed: 19563656]
6. Zhang S, Liu CC, Li W, et al. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. Nucleic Acids Res. 2012; 40:9379–9391. [PubMed: 22879375]

7. Kormaksson M, Booth JG, Figueroa ME, et al. Integrative model-based clustering of microarray methylation and expression data. Ann Appl Stat. 2012; 6:1327–1347.
8. Shen R, Olshen AB, Ladanyi M. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics. 2009; 25:2906–2912. [PubMed: 19759197]
9. Shen R, Mo Q, Schultz N, et al. Integrative subtype discovery in glioblastoma using iCluster. PLoS One. 2012; 7:e35236. [PubMed: 22539962]
10. Shen R, Wang S, Mo Q. Sparse integrative clustering of multiple omics data sets. Ann Appl Stat. 2013; 7:269–294. [PubMed: 24587839]
11. Bartholomew, DJ. Latent Variable Models and Factor Analysis. London: Charles Griffin & Co. Ltd; 1978.
12. Bishop CM, SvensØn M, Williams CKI. GTM: the generative topographic mapping. Neural Comput. 1998; 10:215–234.
13. Dellinger AE, Nixon AB, Pang H. Integrative pathway analysis using graph-based learning with applications to TCGA colon and ovarian data. Cancer Inform. 2014; 13:1–9.
14. Wang W, Baladandayuthapani V, Morris JS, et al. iBAG: integrative Bayesian analysis of high-dimensional multi-platform genomics data. Bioinformatics. 2013; 29:149–159. [PubMed: 23142963]
15. Yang J, Wang X, Kim M, et al. Detection of candidate tumor driver genes using a fully integrated Bayesian approach. Stat Med. 2014; 33:1784–1800. [PubMed: 24347204]
16. Zhao SD, Cai TT, Li H. More powerful genetic association testing via a new statistical framework for integrative genomics. Biometrics. 2014; 70:881–890. [PubMed: 24975802]
17. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B. 1977; 39:1–38.
18. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B. 1996; 58:267–288.
19. Tibshirani R, Walther G. Cluster validation by prediction strength. J Comput Graph Stat. 2013; 14:511–528.
20. de Amorim RC, Hennig C. Recovering the number of clusters in data sets with noise features using feature rescaling factors. Inform Sci. 2015; 324:126–145.
21. Hartigan JA, Wong MA. AS Algorithm 136: a K-means clustering algorithm. J R Stat Soc Ser C Appl Stat. 1979; 28:100–108.
22. Jolliffe, IT. Principal Component Analysis. New York: Springer Verlag; 1986.
23. Wold, H. Path models with latent variables: the NIPALS approach. In: Blalock, HM.Aganbegian, A.Borodkin, FM.Boudon, R., Capecchi, V., editors. Quantitative Sociology: International Perspectives on Mathematical and Statistical Modeling. NewYork: Academic; 1975. p. 307-357.
24. Doledec S, Chessel D. Co-inertia analysis: an alternative method for studying species-environment relationships. Freshw Biol. 1994; 31:277–294.
25. Knapp TR. Canonical correlation analysis: a general parametric significance-testing system. Psychol Bull. 1978; 85:410–416.
26. Hubert L, Arabie P. Comparing partitions. J Classif. 1985; 2:193–218.
27. Verhaak R, Hoadley K, Purdom E, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. Cancer Cell. 2010; 17:98–110. [PubMed: 20129251]
28. Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. J Am Stat Assoc. 1958; 53:457–481.
29. Harrington DP, Fleming TR. A class of rank test procedures for censored survival data. Biometrika. 1982; 69:553–566.

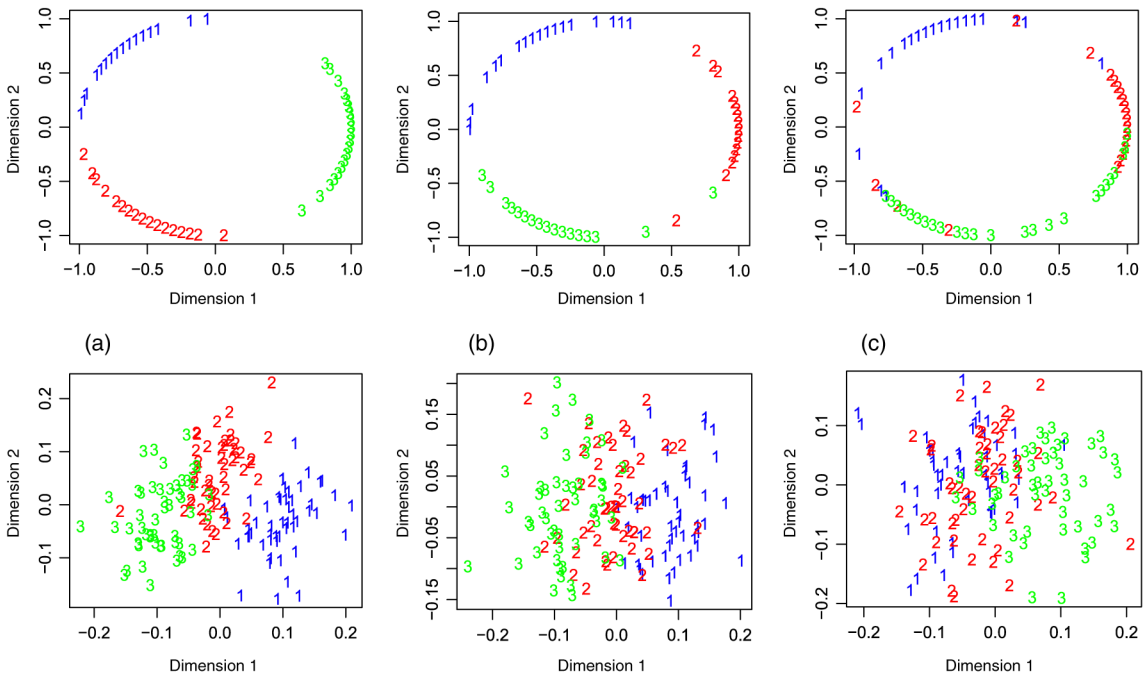

Fig. 1.

Plots of the first two dimensions after dimension reduction by both methods compared for each case. Panels (a–c) and Panels (d–f) correspond to Cases 1–3, respectively. Here, irGTM, iCluster denote the proposed method (irGTM$_1$) and iCluster of ref. [8], respectively. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

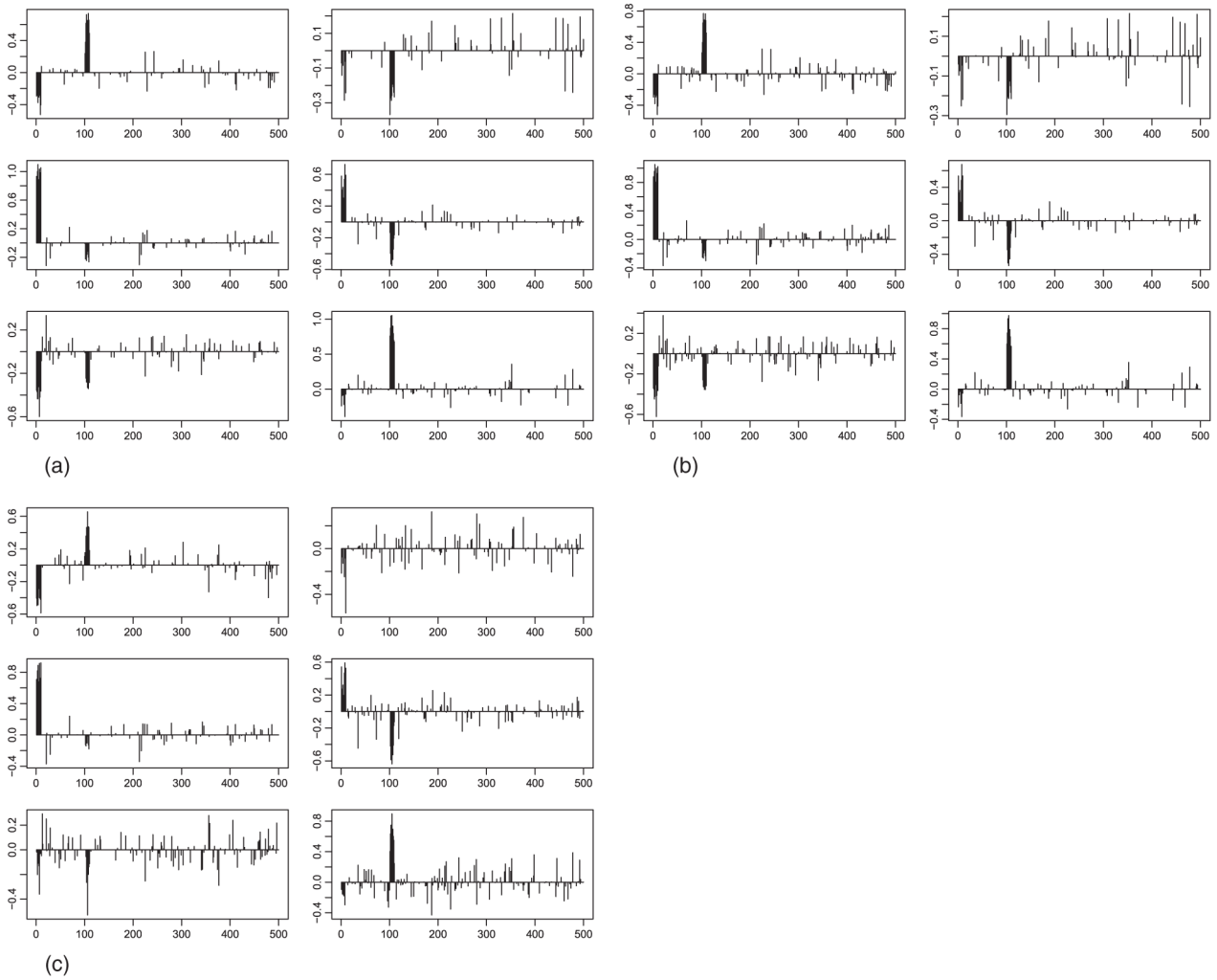

Fig. 2.

Plots of the estimates of parameters in irGTM$_1$. Panels (a–c) correspond to Cases 1–3. For each panel, the (k, s)-th entry is the plot of $W_k^{(s)}$ (the kth column of $\boldsymbol{W}^{(s)}$) for $k \in \{1, \cdots, K = 3\}$ and $s \in \{1, \cdots, S = 2\}$.

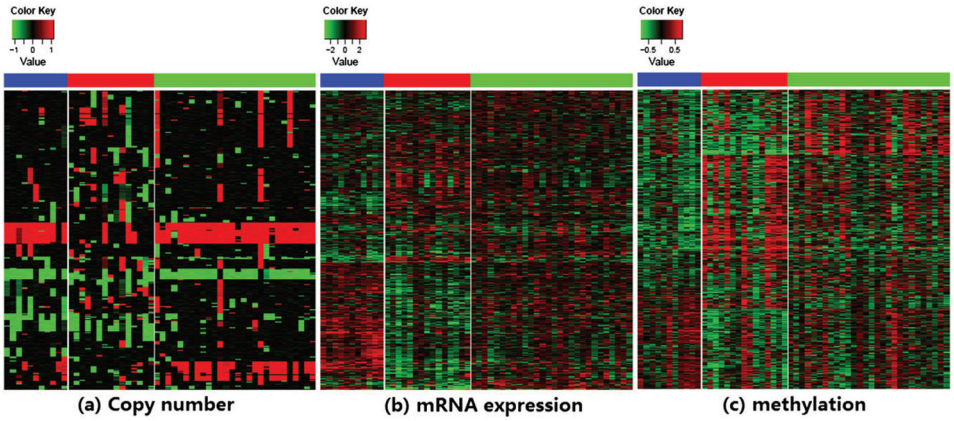

Fig. 3.

Panels (a–c) are heat maps of the DNA copy number data, the mRNA expression data and the methylation data of the 55 GBM patients from The Cancer Genome Atlas (TCGA) respectively, where the columns (tumors) are arranged by clusters identified by irGTM$_1$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

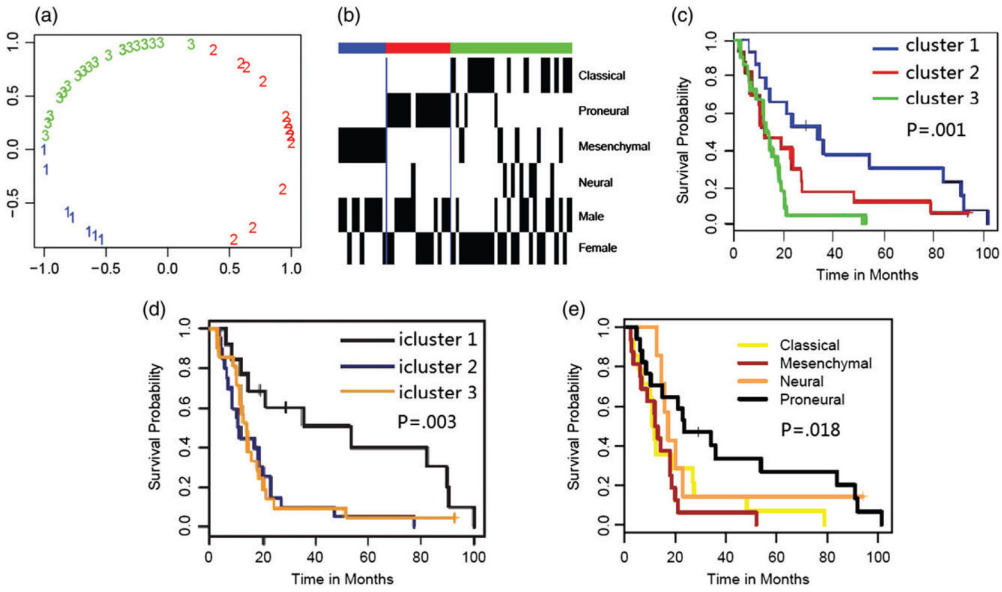

Fig. 4.

Panels (a–e) are based on the DNA copy number data, the mRNA expression data and the methylation data of the 55 GBM patients from TCGA, where P denotes the p-value of the Mantel-Haenszel test. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 1

Sample means (SD in parentheses) of Rand index (Rand) and adjusted Rand index (aRand) between the clustering assignment by each method and the true assignment as well as the run time (RT) for the three cases based on 100 repetitions. Here, irGTM$_1$, irGTM$_2$, iCluster, PCA NI, PLS, CIA, and CCA denote the proposed method with different orders of data types, iCluster of ref. [8], naive integration of PCA, partial least squares regression of ref. [23], co-inertia analysis of ref. [24], and canonical correlation analysis of [25].

Case	n, p, μ	Method	Rand	aRand	RT (min)
1	$n = 150$	irGTM$_1$	**0.987** (0.012)	**0.972** (0.028)	12.37 (0.036)
		irGTM$_2$	**0.988** (0.011)	**0.974** (0.022)	12.32 (0.032)
	$p^{(1)} = p^{(2)}$	iCluster	0.978 (0.018)	0.952 (0.042)	152.5 (10.44)
	$= 500$	PCA NI	0.923 (0.024)	0.826 (0.056)	0.004 (0.001)
	$\mu = 1.5$	PLS	0.762 (0.063)	0.464 (0.143)	0.027 (0.008)
		CIA	0.888 (0.030)	0.746 (0.069)	0.008 (0.002)
		CCA	0.954 (0.065)	0.896 (0.147)	0.007 (0.001)
2	$n = 150$	irGTM$_1$	**0.979** (0.011)	**0.954** (0.026)	12.26 (0.260)
		irGTM$_2$	**0.981** (0.013)	**0.957** (0.029)	12.28 (0.281)
	$p^{(1)} = p^{(2)}$	iCluster	.946 (0.041)	0.877 (0.092)	152.0 (9.992)
	$= 500$	PCA NI	0.756 (0.047)	0.450 (0.105)	0.006 (0.000)
	$\mu = 1.3$	PLS	0.722 (0.054)	0.373 (0.120)	0.025 (0.000)
		CIA	0.727 (0.050)	0.387 (0.112)	0.008 (0.000)
		CCA	0.901 (0.079)	0.778 (0.178)	0.006 (0.000)
3	$n = 150$	irGTM$_1$	**0.938** (0.019)	**0.855** (0.043)	12.28 (0.308)
		irGTM$_2$	**0.936** (0.023)	**0.856** (0.051)	12.27 (0.320)
	$p^{(1)} = p^{(2)}$	iCluster	0.843 (0.085)	0.647 (0.191)	84.96 (16.99)
	$= 500$	PCA NI	0.677 (0.044)	0.272 (0.100)	0.005 (0.000)
	$\mu = 1.1$	PLS	0.657 (0.030)	0.229 (0.067)	0.018 (0.000)
		CIA	0.656 (0.032)	0.227 (0.072)	0.005 (0.000)
		CCA	0.792 (0.101)	0.531 (0.226)	0.005 (0.000)

Table 2

Running time (in minutes), Rand and adjusted Rand index results of Cases 1–3 by using irGTM$_1$ with $\phi(\boldsymbol{Z})$ = arccos $\phi(\boldsymbol{Z})=\arccos\left(\frac{\boldsymbol{Z}^T\mu_k}{\|\boldsymbol{Z}\|\|\mu_k\|}\right)$.

	Rand	aRand	RT (min)
Case 1	0.984 (0.012)	0.964 (0.027)	8.780 (0.422)
Case 2	0.969 (0.018)	0.931 (0.042)	8.634 (0.416)
Case 3	0.920 (0.048)	0.819 (0.110)	8.792 (0.425)

Table 3

Running time (in minutes), Rand and adjusted Rand index results of Cases 1–3 by using irGTM$_1$ with an alternative choice of latent space (three-dimensional space) and cluster centers.

	Rand	aRand	RT (min)
Case 1	0.988 (0.010)	0.974 (0.023)	24.57 (0.206)
Case 2	0.973 (0.027)	0.939 (0.061)	24.59 (0.170)
Case 3	0.933 (0.061)	0.849 (0.139)	24.56 (0.186)

Table 4

Running time (in minutes), Rand and adjusted Rand index results of Cases 1–3 by using irGTM with an alternative choice of basis functions of latent variables.

	Rand	aRand	RT (min)
Case 1	0.986 (0.023)	0.970 (0.051)	5.873 (0.046)
Case 2	0.967 (0.045)	0.927 (0.101)	5.953 (0.079)
Case 3	0.926 (0.076)	0.833 (0.171)	5.976 (0.078)

Table 5

Running time (in minutes), Rand and adjusted Rand index results of Cases 1–3 by using irGTM$_1$ with alternative choices of the scale of the proposed basis function.

	δ	Rand	aRand
Case 1	0.5	0.990 (0.011)	0.978 (0.022)
	0.75	0.990 (0.011)	0.979 (0.026)
	1.5	0.976 (0.016)	0.946 (0.036)
Case 2	0.5	0.981 (0.013)	0.957 (0.031)
	0.75	0.982 (0.045)	0.959 (0.101)
	1.5	0.923 (0.078)	0.828 (0.175)
Case 3	0.5	.956 (0.018)	0.900 (0.042)
	0.75	.958 (0.020)	0.906 (0.045)
	1.5	.849 (0.099)	0.661 (0.223)