Thompson River University

Review of iClusterVB

A Fast Integrative Clustering and Feature Selection Approach for High-Dimensional Data

Hadiseh Azadehyaei (T00726889)
Feng Gu (T00751197)

April 2025

Abstract

The `iClusterVB` framework was presented in our data science seminar class and immediately captured our interest. As a result, we thoroughly reviewed the algorithm and implemented it on a simulated dataset to better understand the process. Subsequently, we applied the method to a breast cancer dataset. This technique provides a principled approach to integrative clustering by combining finite mixture models, variational Bayesian inference, and embedded feature selection. This methodology is particularly well-suited for high-dimensional, multi-type data commonly encountered in genomics and biomedical research. At its core, `iClusterVB` models the data as arising from a mixture of latent clusters, each described by its own distribution. Feature relevance is determined via a binary latent indicator, allowing the model to automatically select informative features while ignoring noise.

The framework supports continuous, categorical, and count data through appropriate likelihood functions, and leverages conjugate priors to enable efficient posterior updates. Variational inference is employed to approximate the intractable posterior distributions, replacing sampling-based methods with an optimization-driven approach that minimizes the Kullback-Leibler (KL) divergence. The optimization maximizes the Evidence Lower Bound (ELBO), ensuring both fidelity to the data and model parsimony.

In the breast cancer analysis, we applied `iClusterVB` to the TCGA-BRCA dataset, where the algorithm determined an optimal clustering solution with five distinct clusters among 348 individuals. These clusters were identified based on gene expression, DNA methylation, and miRNA expression data. The framework successfully selected 488 features from gene expression data, 437 from DNA methylation, and 193 from miRNA expression, each surpassing the posterior inclusion probability threshold of 0.5. The ELBO value of $-57,406,131.13$ after 100 iterations indicated convergence, reflecting the model's ability to extract biologically relevant patterns from the high-dimensional omics data.

Overall, after reviewing the whole `iClusterVB` algorithm and its application to both simulated data and the breast cancer TCGA data, we found that it offers a scalable, interpretable, and flexible solution for model-based clustering in complex multi-omic and high-dimensional settings. This is demonstrated in the breast cancer analysis, where it successfully uncovered distinct molecular subtypes with meaningful biological implications.

1 Introduction

Clustering analysis plays a fundamental role in revealing hidden patterns in complex biological data. It involves grouping similar data points such that those within a cluster are more similar to each other than to those in other clusters. In the field of bioinformatics, clustering has proven instrumental in tasks such as disease subtyping, biomarker discovery, and patient stratification—essential steps toward personalized medicine. Notable successes include the identification of distinct subtypes in diseases such as asthma, Parkinson's disease, and various cancers, where clustering has enabled deeper biological insights and guided downstream predictive modeling.

However, the increasing complexity and volume of biomedical data have introduced new challenges that traditional clustering methods—such as k-means or hierarchical clustering—are often not well-equipped to address. Modern datasets, particularly in genomics and cancer research, are frequently multimodal, combining data from gene expression, DNA methylation, miRNA expression, mutations, and more. These data types vary in structure—continuous, binary, categorical—and span tens of thousands of features, only a fraction of which are typically informative. This high dimensionality and heterogeneity demand advanced statistical methods capable of integrative analysis and feature selection across multiple data views.

Integrative clustering methods have emerged as a powerful solution to these challenges by simultaneously analyzing multiple data types to identify robust, biologically meaningful subtypes. Among the most promising approaches are Bayesian frameworks, which incorporate prior knowledge to improve inference quality. While traditional Bayesian methods often rely on computationally intensive techniques such as Markov Chain Monte Carlo (MCMC), more recent developments like Variational Bayes (VB) provide faster and more scalable alternatives by transforming the inference task into an optimization problem.

The iClusterVB R package, introduced by Alnajjar and Lu (2024), leverages a variational Bayesian approach to enable efficient integrative clustering and feature selection in high-dimensional, mixed-type data. The package builds upon the strengths of earlier tools like iClusterPlus and iClusterBayes while addressing their computational limitations. Key features of iClusterVB include support for continuous, discrete, and categorical data types; automatic selection of the optimal number of clusters; and a scalable algorithm well-suited for modern multi-omics applications.

In this review, we aim to assess the practical utility and generalizability of the iClusterVB method by applying it to simulated data.

2 iClusterVB methodology and practical steps

2.1 iClusterVB methodology

The `iClusterVB` framework is built on a Bayesian model-based clustering approach using a finite mixture model, variational inference, and embedded feature selection.

2.2 Finite Mixture Model

A finite mixture model is a probabilistic model that represents a population as a mixture of several subpopulations, each described by its own probability distribution. It is widely used in clustering and density estimation. [1]

Let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ be the p-dimensional

feature vector for individual i, and let z_i be the cluster assignment. The likelihood of the data is modeled using a finite mixture model:

$$P(\mathbf{x}_i \mid \boldsymbol{\pi}, \Theta) = \sum_{k=1}^{K} \pi_k P(\mathbf{x}_i \mid \phi_k)$$

Assuming conditional independence of features within each cluster:

$$P(\mathbf{x}_i \mid \phi_k) = \prod_{j=1}^{p} P(x_{ij} \mid \phi_{kj})$$

where ϕ_k are parameters for the k-th cluster, and π_k are mixing proportions such that $\sum_k \pi_k = 1$.

2.3 Feature Selection

To select informative features, the model includes a latent indicator $\gamma_{ij} \in \{0, 1\}$ that controls whether feature j is relevant for clustering:

$$P(x_{ij} \mid \gamma_{ij}, z_i = k) = \begin{cases} P(x_{ij} \mid \theta_{kj}), & \text{if } \gamma_{ij} = 1 \\ P(x_{ij} \mid \eta_j), & \text{if } \gamma_{ij} = 0 \end{cases}$$

The marginal likelihood becomes:

$$P(\mathbf{x}_i \mid z_i = k) = \prod_{j=1}^{p} \left[\omega_j P(x_{ij} \mid \theta_{kj}) + (1 - \omega_j) P(x_{ij} \mid \eta_j)\right]$$

[1] For more details, refer to the Wikipedia article on finite mixture models.

where ω_j is the inclusion probability of feature j.

Distributions for different types of feature.

	$p(x_{ij} \mid \boldsymbol{\vartheta}_{kj})$	$p(x_{ij} \mid \boldsymbol{\eta}_j)$
Continuous features	$\mathcal{N}(\mu_{kj}, \sigma^2_{kj})$	$\mathcal{N}(\iota_j, \epsilon^2_j)$
Categorical features	Multinomial$(1; \theta_{kj1}, \ldots, \theta_{kjM_j})$	Multinomial$(1; \zeta_{j1}, \ldots, \zeta_{jM_j})$
Count features	Poisson(λ_{kj})	Poisson(ν_j)

Table 1: Distributions used for different feature types under cluster-specific and marginal models.

2.4 Prior Distributions

Priors are imposed to enable Bayesian inference, leveraging the conjugate prior method for computational efficiency and analytical tractability. Conjugate priors ensure that the posterior distribution belongs to the same family as the prior, simplifying updates during inference.

Likelihood Function	**Parameter**	**Prior Distribution**	**Posterior Distribution**
$P(\mathbf{x}_i \mid \boldsymbol{\pi})$	$\boldsymbol{\pi}$	Dirichlet(α_0)	Dirichlet$(\alpha_0 + \text{counts})$
$P(x_{ij} \mid \mu_{kj}, \sigma^2_{kj})$	μ_{kj}	$\mathcal{N}(\mu_0, s^2_0)$	$\mathcal{N}(\mu_n, s^2_n)$
$P(x_{ij} \mid \sigma^2_{kj})$	σ^2_{kj}	IG(a_0, b_0)	IG(a_n, b_n)
$P(x_{ij} \mid \boldsymbol{\theta}_{kj})$	$\boldsymbol{\theta}_{kj}$	Dirichlet$(\boldsymbol{\kappa}_{kj})$	Dirichlet$(\boldsymbol{\kappa}_{kj} + \text{counts})$
$P(x_{ij} \mid \lambda_{kj})$	λ_{kj}	Gamma(c_0, d_0)	Gamma(c_n, d_n)

Table 2: Conjugate Priors and Corresponding Posterior Distributions with Likelihood Functions and Parameters

- **Mixing weights:** $\boldsymbol{\pi} \sim$ Dirichlet(α_0) ensures that the mixing proportions are non-negative and sum to 1. The posterior updates naturally with observed cluster counts.
- **Gaussian features:** Conjugate priors for mean ($\mathcal{N}$) and variance (IG) allow efficient updates based on sufficient statistics (e.g., sample mean and variance).
- **Categorical features:** Dirichlet priors for multinomial parameters $\boldsymbol{\theta}_{kj}$ simplify posterior updates with observed category counts.
- **Count features:** Gamma priors for Poisson rates λ_{kj} provide a flexible framework for modeling count data, with posterior updates driven by observed counts.

2.5 Variational Inference

To approximate the posterior distribution $P(\beta \mid \mathbf{X})$, iClusterVB uses a method called variational inference. Instead of relying on computationally expensive sampling methods, it turns the problem into an optimization task.

The goal is to find a simpler distribution $Q(\beta)$ that is as close as possible to the true posterior $P(\beta \mid \mathbf{X})$. This is done by minimizing a measure called KL divergence, which quantifies the difference between the two distributions:

$$\mathrm{KL}(Q(\beta) \parallel P(\beta \mid \mathbf{X})) = \int Q(\beta) \log \frac{Q(\beta)}{P(\beta \mid \mathbf{X})} \, d\beta$$

In practice, this is equivalent to maximizing a quantity called the Evidence Lower Bound (ELBO). The ELBO is defined as:

$$\mathrm{ELBO} = \mathbb{E}_{Q(\beta)}[\log P(\mathbf{X}, \beta)] - \mathbb{E}_{Q(\beta)}[\log Q(\beta)]$$

The ELBO balances two terms: how well the model explains the data and how simple the approximating distribution is.

Additionally, iClusterVB can automatically adjust the number of clusters by checking the estimated mixing proportions $\hat{\pi}_k$. If a cluster's proportion is too small (e.g., below a threshold like 0.01), that cluster is removed to simplify the model.

3 iClusterVB Implementation on simulation data

3.1 Data source and data description

In this section, we demonstrate the functionality of `iClusterVB` for integrative clustering and feature selection using a simulated multi-omic dataset. This simulation is designed to closely mimic the structure of real-world biological data typically encountered in genomics and precision medicine studies.

The simulated dataset includes data from $N = 240$ individuals, each with measurements across $R = 4$ different data types. Specifically, two of these data types represent **continuous variables**—such as gene or mRNA expression levels—while the third data type consists of **count data**, resembling DNA copy number variations. The fourth data type is **binary**, capturing the presence or absence of mutations or other genomic aberrations. This combination of data types reflects a common setup in integrative genomic analyses, where multiple omic layers are collected for each subject.

We assume that the underlying true structure of the data consists of $K = 4$ distinct clusters, corresponding to four biological subgroups. These clusters are **balanced in size**, meaning each cluster contains 25% of the individuals ($\pi_1 = \pi_2 = \pi_3 = \pi_4 = 0.25$).

Each data type contains $p_r = 500$ features, for a total of $p = 2000$ features across all data types. Among these, **only 10%** (i.e., 50 features per data type) are informative and contribute to the clustering structure. These informative features are also referred to as **relevant or discriminative features**, as they capture the variations that distinguish between clusters. The remaining 90% of the features in each data type serve as **irrelevant noise features**, which do not contribute to the clustering and are expected to be filtered out during the feature selection process.

The distribution of these useful features across the four clusters is structured, and the table below summarizes how the relevant features are assigned within each cluster and data type.

Table 3: Distribution of relevant and noise features across clusters in each data view of the simulated data

Data View	**Cluster**	**Distribution**
1 (Continuous)	Cluster 1	$\mathcal{N}(10, 1)$ (Relevant)
	Cluster 2	$\mathcal{N}(5, 1)$ (Relevant)
	Cluster 3	$\mathcal{N}(-5, 1)$ (Relevant)
	Cluster 4	$\mathcal{N}(-10, 1)$ (Relevant)
	-	$\mathcal{N}(0, 1)$ (Noise)
2 (Continuous)	Cluster 1	$\mathcal{N}(-10, 1)$ (Relevant)
	Cluster 2	$\mathcal{N}(-5, 1)$ (Relevant)
	Cluster 3	$\mathcal{N}(5, 1)$ (Relevant)
	Cluster 4	$\mathcal{N}(10, 1)$ (Relevant)
	-	$\mathcal{N}(0, 1)$ (Noise)
3 (Binary)	Cluster 1	Bernoulli(0.05) (Relevant)
	Cluster 2	Bernoulli(0.2) (Relevant)
	Cluster 3	Bernoulli(0.4) (Relevant)
	Cluster 4	Bernoulli(0.6) (Relevant)
	-	Bernoulli(0.1) (Noise)
4 (Count)	Cluster 1	Poisson(50) (Relevant)
	Cluster 2	Poisson(35) (Relevant)
	Cluster 3	Poisson(20) (Relevant)
	Cluster 4	Poisson(10) (Relevant)
	-	Poisson(2) (Noise)

3.2 Exploratory Data Analysis (EDA)

To identify and visualize the valid normal distributions from the simulated dataset, we performed the following sampling and testing operations:

- **Random Sampling:** We randomly sampled 30 variables from each of the three different distributions (normal, poisson, and multinomial), resulting in a total of 90 variables for testing.
- **Statistical Testing:** For each sampled variable, we conducted the corresponding statistical tests:
  - A normality test (e.g., Shapiro-Wilk test) for variables assumed to follow a normal distribution.
  - A Poisson test for variables assumed to follow a Poisson distribution.
  - A multinomial test for variables assumed to follow a multinomial distribution.
- **Validation and Visualization:** Variables that passed their respective tests were visualized to determine whether they represent truly valid features or are merely low-probability events arising from noise distributions.

This process helps to filter out irrelevant noise features and retain only the informative variables that contribute to the clustering structure.

Figure 1 displays histograms of variables simulated from normal distributions. Variables with titles including "(Noise)" correspond to noise components sampled from $\mathcal{N}(0, 1)$, while those without such a label represent relevant features derived from mixtures of $\mathcal{N}(10, 1)$, $\mathcal{N}(5, 1)$, $\mathcal{N}(-5, 1)$, and $\mathcal{N}(-10, 1)$. These relevant variables show clear multimodal patterns, indicating the presence of structured information. In contrast, the noise variables exhibit unimodal, symmetric distributions centered around zero, lacking informative structure. The absence of "(Noise)" in the title marks the variables that passed the normality test and were deemed statistically significant for downstream modeling.

Figure 2 shows histograms of variables generated from Poisson distributions with different rate parameters λ. Variables labeled with multiple λ values (e.g., $\lambda = 50, 35, 20, 10$) represent informative features composed of mixed Poisson signals. These histograms exhibit broader, multi-modal or skewed distributions, suggesting meaningful variability across groups, which is useful for clustering tasks. In contrast, variables marked as "(Noise)"—such as those with $\lambda = 2$ only—display narrow, concentrated distributions around small integer values. These noise variables do not carry discriminative structure and are thus considered irrelevant for clustering.

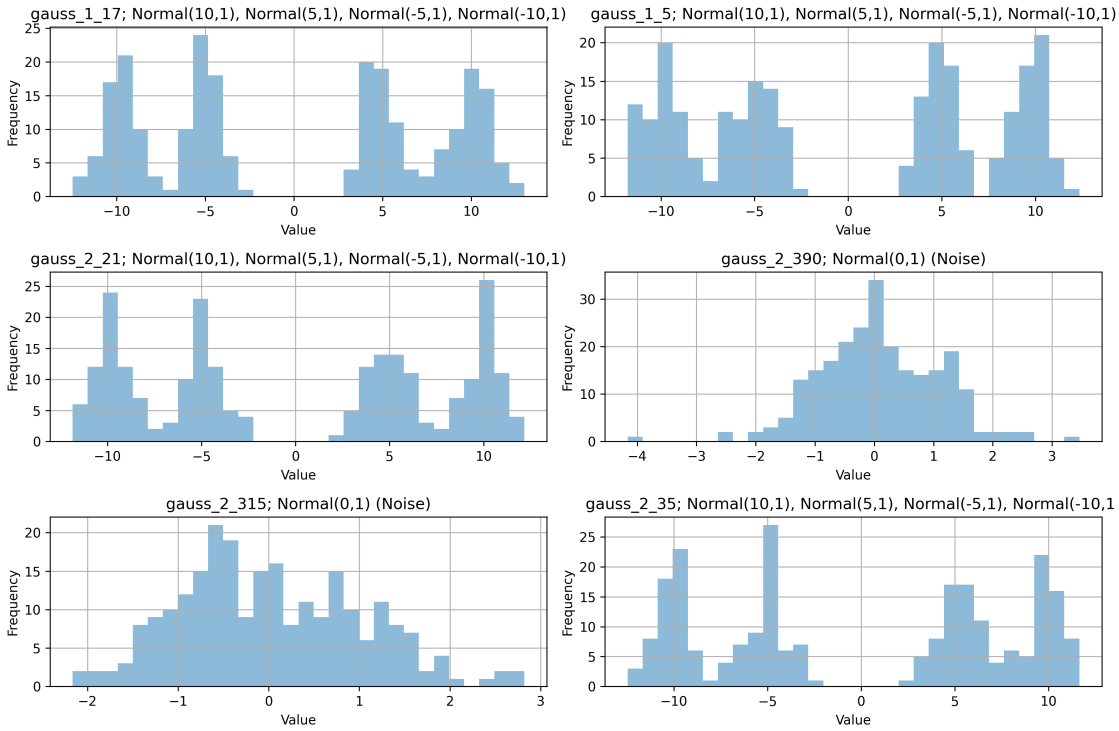

Figure 1: Histograms of normal distributions that pass the normality test(correspond to table 9)

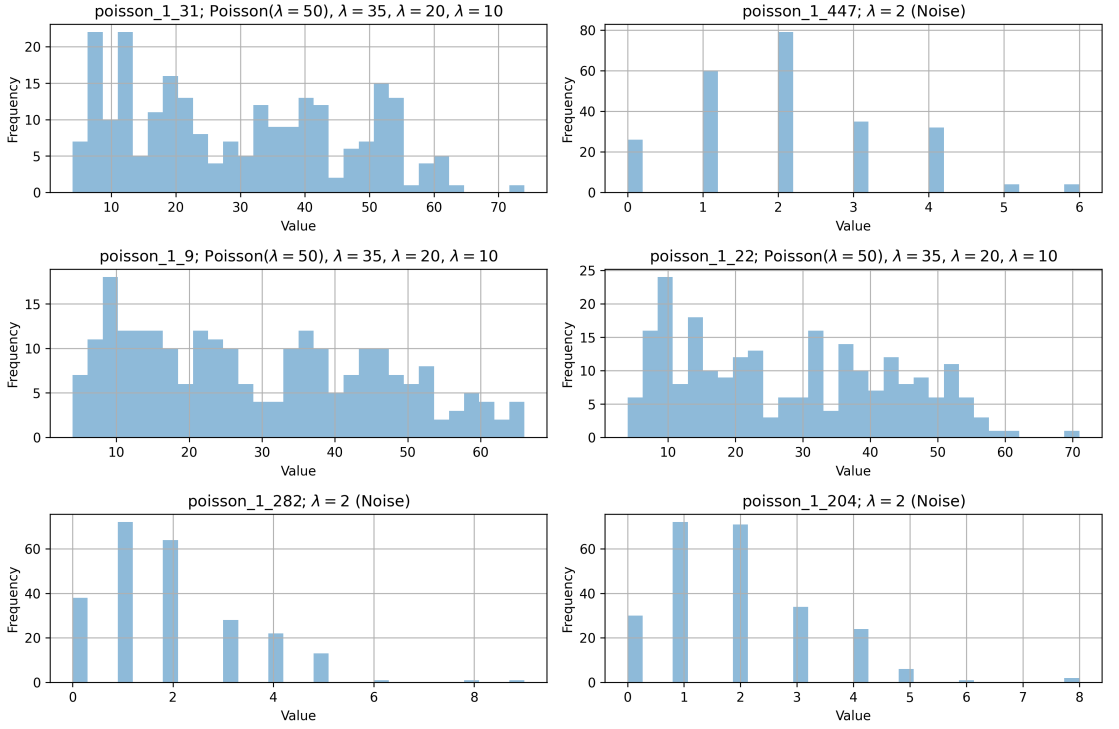

Figure 2: Histograms of normal distributions that pass the poission test (corresponding to table 10)

3.3 Necessity of testing in EDA for iClusterVB algorithm

The testing operations described above (numerical information are stored in tables 9, 10, 11 in appendix) demonstrate the effectiveness of statistical tests in identifying suitable variables for downstream analysis. By applying normality, Poisson, and multinomial tests, we were able to filter out irrelevant noise features and retain only the informative variables that exhibit meaningful patterns or variability. These selected variables are well-suited for integrative clustering and feature selection tasks, making them ideal candidates for input into the `iClusterVB` algorithm. This ensures that the algorithm operates on high-quality data, enhancing its ability to uncover biologically meaningful clusters and informative features.

3.4 iClusterVB Results on simulation data

In this section, we show how the **iClusterVB** method works using simulated data. The data includes 240 individuals and 4 different types of data views. These data types are often found in genomics studies:

- Two continuous views (like gene expression)
- One binary view (like mutation: yes or no)
- One count view (like DNA copy number)

Each view has 500 features, so in total we have 2000 features. But only 10% of features (50 per view) are useful for clustering. The rest are just noise (not important).

The true number of clusters is 4, and each cluster has the same number of individuals (60 each).

We combined these datasets into a list and specified their types using a vector. Before running the model, we changed the 0 values in the binary data to 2, because the algorithm does not accept zeros. Then, we used the iClusterVB function to run the clustering model with a maximum of 8 clusters. The model correctly found 4 clusters. After that, we used summary to check the results and piplot and chmap to visualize selected features and clusters.

The clustering algorithm identified 4 clusters from the simulated dataset, each containing 60 individuals, for a total of 240, despite a maximum of 6 clusters being specified.

Variable selection identified 57 and 58 relevant variables in the two Gaussian views, 63 in the Multinomial view, and 68 in the Poisson view (out of 500 per view), indicating that all data types contributed meaningful signals, with slightly stronger effects from the Multinomial and Poisson views.

Table 4: Summary of Clustering and Variable Selection Results

Category	**Description**	**Value**
Clustering Hyper-Parameters/Result		
Number of individuals	Total number of observations	240
Maximum clusters specified	User-defined input	6
Number of clusters found	Determined by algorithm	4
Individuals per cluster	Cluster sizes (equal)	60
Variable Selection by View (Threshold: 0.5)		
View 1 (Gaussian)	Variables selected	57 out of 500
View 2 (Gaussian)	Variables selected	58 out of 500
View 3 (Multinomial)	Variables selected	63 out of 500
View 4 (Poisson)	Variables selected	68 out of 500

Table 5: Cross-tabulation of Predicted vs. True Cluster Memberships(column 2 to 5 are true clusters in data)

Predicted Cluster	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
1	0	0	60	0
2	0	60	0	0
3	60	0	0	0
6	0	0	0	60

Table 5 presents the cross-tabulation between the predicted cluster labels and the true underlying cluster assignments. Each predicted cluster perfectly matches one of the true clusters without any misclassification. Specifically, predicted cluster 1 matches true cluster 3, cluster 2 matches true cluster 2, cluster 3 matches true cluster 1, and cluster 6 corresponds exactly to true cluster 4. This result indicates a perfect clustering performance, with all 240 individuals correctly assigned to their respective groups.

Figure 3 illustrates the posterior probability of variable inclusion across the four different data views: (A) and (B) correspond to the two Gaussian views, (C) corresponds to the Multinomial view, and (D) to the Poisson view. In each subplot, the radial axis represents the probability of inclusion for individual variables, with higher values indicating stronger evidence of variable relevance. The angular separation reflects the clustering structure among variables.

The color gradient encodes the ρ value, which governs the correlation or dependence strength across features. Bright blue regions (high ρ) indicate confidently selected variables, while darker tones indicate low inclusion probability or noise. Across all views, the most informative variables are highlighted clearly with high inclusion probabilities concentrated

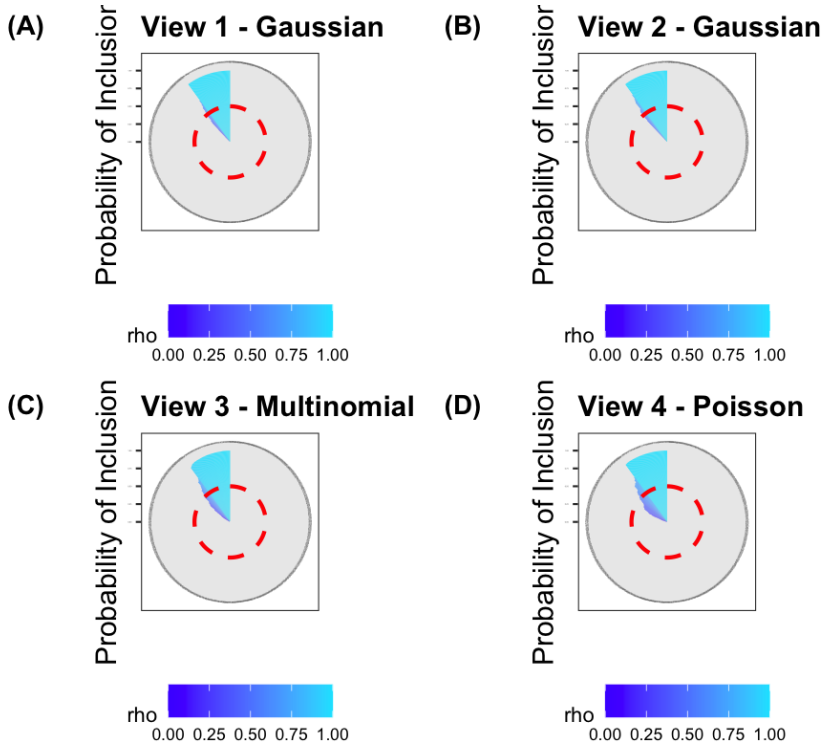

Figure 3: PIP plot showing the posterior inclusion probabilities across all data views.

in narrow angular bands, consistent with the true signal structure of the simulated dataset.

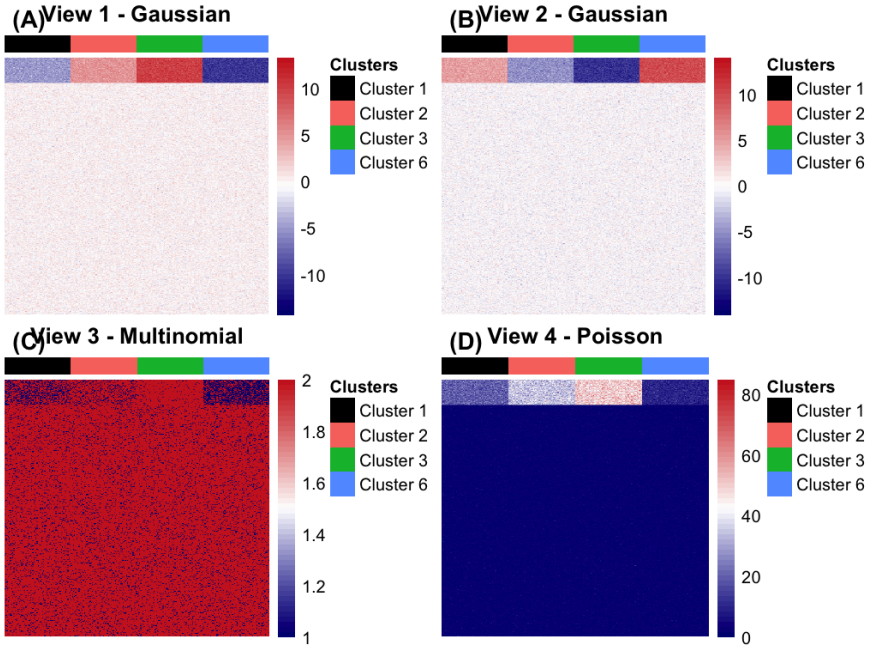

Figure 4: Cluster heatmap showing the clustering structure across all data views.

Figure 4 shows heatmaps of the four data views used for clustering. Clear block patterns in Views 1 and 2 (Gaussian) indicate that these features distinguish clusters well. View 3 (Multinomial) also displays cluster-specific structure in the upper rows. While View 4 (Poisson) is mostly sparse, the top block shows variation across clusters, suggesting some informative features. Together, these heatmaps validate that each view contributes to identifying distinct clusters.

4 iClusterVB Implementation on Breast Cancer

In this section, we transition from working with simulation data to applying our methodology to a more realistic dataset.

Specifically, we utilize the Breast Cancer dataset[2] obtained from Kaggle. This dataset provides a practical context to evaluate the performance and applicability of the iClusterVB implementation in addressing real-world challenges.

There are no missing values in the dataset, and all features are numeric(except the target variable(M & B)), which provide us convenient conditions for applying the iClusterVB method.

Before taking statistical testing, we did some data preprocessing. We encoded the target variable and split the data based on the target variable, got two subsets of data, one for M and one for B. After checking the distributions of the two data sets, we found there were heavy skewness and outliers in the data, therefore we make a simple logarithm transformation on the data [3].

4.1 Distribution Testing

By normality testing, we found that most variables are not normally distributed, only four variables are normally distributed, which are: *area_se, texture_worst, concave points_worst, symmetry_worst*.

Hence, we will take these four variables as valid normally distributed features as one of the inputs for the iClusterVB method.

We also did Poisson testing on the data, and we found that most variables are not Poisson distributed, only one variable *texture_se* is Poisson distributed.

Other variables that are not shown in the table are not suitable for the Poission test, and we assumed they are not Poisson distributed.

To multinomial(bernoulli) test, we found that all variables are not interget-valued, hence they are not suitable for multinomial test. Therefore, we will not take multinomial test in this case.

Table 7 lists the variables selected for the iClusterVB analysis. The Gaussian variables were chosen based on their normality test results, while the Poisson variable was selected based on the Poisson test. These variables serve as inputs for the iClusterVB method,

[2]**This Breast Cancer data set is totally different from the data set used in section 6, therefore we name it with 'typical Cancer data' in title to avoid confusion.**

[3]In later practice, other transfomration can be used here, taking log-transfomration is only for convenience of demonstration here

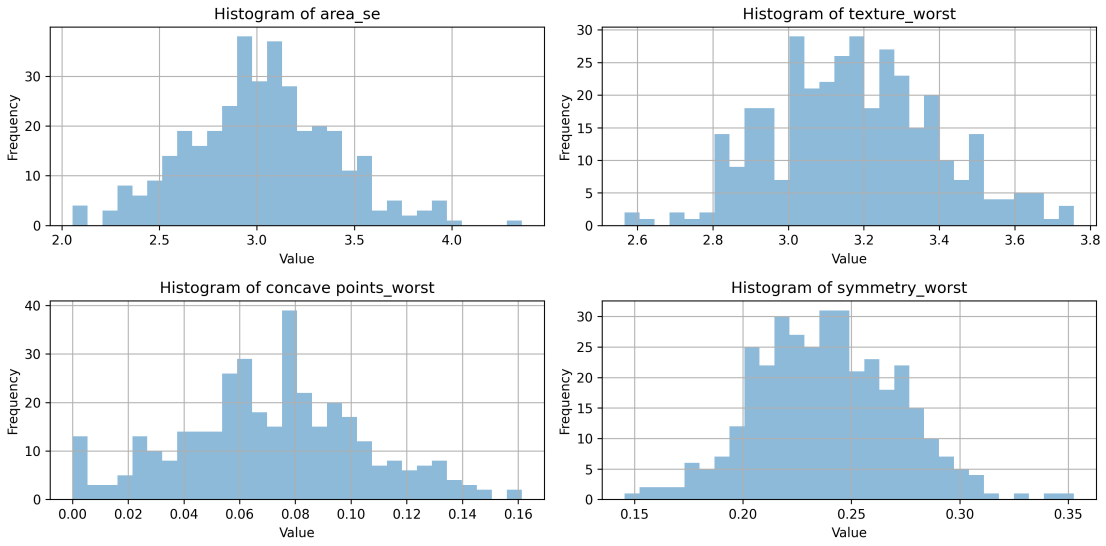

Figure 5: Histograms of normal distributions that pass the normality test(corresponding to table 12)

Table 6: Chi-square test for all variables in the breast cancer dataset

index	Variable	Chi2 Statistic	Poisson-p-value
3	texture_mean	25.2086	0.0000
4	perimeter_mean	334.8591	0.0000
5	area_mean	232.2412	0.0000
13	texture_se	1.5309	0.2160
14	perimeter_se	63.7165	0.0000
15	area_se	108.8869	0.0000
22	radius_worst	235.0098	0.0000
23	texture_worst	137.9469	0.0000
24	perimeter_worst	353.2600	0.0000
25	area_worst	354.8223	0.0000

Table 7: Variables Selected for iClusterVB Analysis

Variable Type	Selected Variables
Gaussian (Normal)	area_se, texture_worst, concave points_worst, symmetry_worst
Poisson	texture_se

representing distinct data views to capture meaningful patterns and relationships in the dataset.

4.2 iClusterVB Results on Cancer Subset

The clustering algorithm was applied to a real subset of the breast cancer dataset, yielding the following results:

Table 8: Clustering Summary for Breast Cancer Subset

Metric	**Result**
Total number of individuals	357
Maximum number of clusters (user input)	6
Number of clusters determined	6
Variables selected (posterior inclusion probability 0.5)	
View 1 - Gaussian	2 out of 4
View 2 - Poisson	2 out of 2

Cluster Membership Distribution

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
57	55	80	46	98	21

Table 8 presents the clustering summary for a subset of the breast cancer dataset. The algorithm successfully identified 6 clusters from 357 individuals, matching the user-specified maximum number of clusters. Cluster sizes ranged from 21 to 98. Variable selection results show that 2 out of 4 Gaussian variables and all 2 Poisson variables were selected based on a posterior inclusion probability threshold of 0.5, indicating their importance in distinguishing clusters across the two views.

Figure 6 shows the clustering heatmaps for the breast cancer dataset based on two data views: (A) Gaussian and (B) Poisson. Each column corresponds to an individual, and rows represent variables. The color bars at the top denote the cluster membership of each individual. In both views, clear block patterns are observed, indicating distinct variable activity across different clusters. This separation suggests that the clustering algorithm effectively identified subgroups with different underlying characteristics in both Gaussian and Poisson features.

4.3 iClusterVB Implementation on TCGA data set

The second dataset was downloaded from `https://portal.gdc.cancer.gov/projects/TCGA-BRCA`. It includes multi-omic data, including gene expression, DNA methylation, and miRNA expression for 348 individuals diagnosed with breast cancer. The analysis was conducted using the `iClusterVB` framework, which leverages variational Bayesian inference

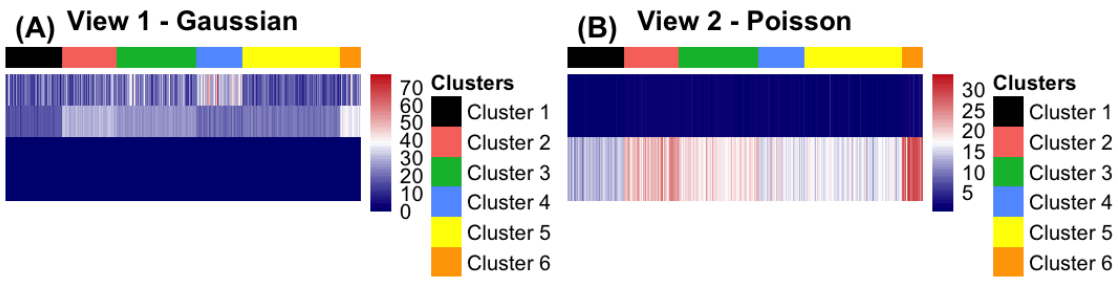

Figure 6: Heatmap of the clustering results on breast cancer data

for integrative clustering. The R code used in the analysis involved loading the necessary multi-omic data sets, specifying the clustering parameters (such as the number of clusters and feature selection method), and fitting the model using the `iClusterBayes` function. The model's posterior distributions were approximated using variational inference, and the results were visualized through heatmaps and posterior inclusion probability plots to identify informative features and cluster memberships. The `iClusterVB` framework effectively captured the heterogeneity within the dataset, revealing biologically meaningful subtypes of breast cancer based on molecular data.

At this stage, we perform exploratory data analysis (EDA) to better understand the breast cancer dataset.

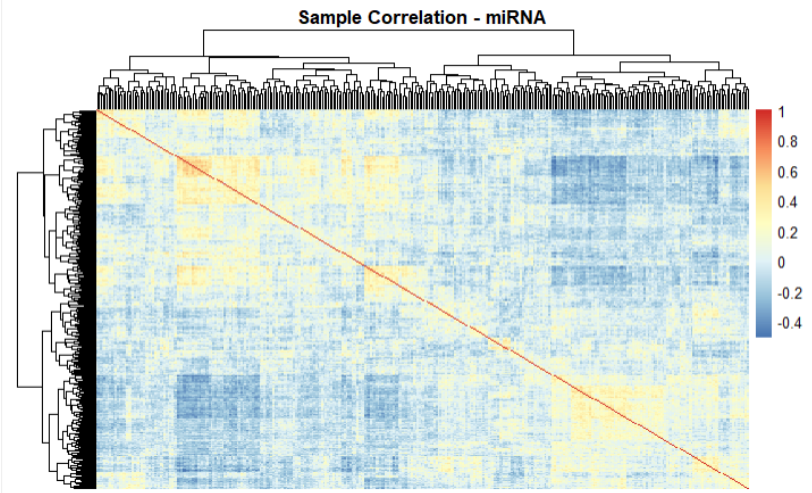

Figure 7: Sample Correlation - miRNA

The heatmaps "Sample Correlation – miRNA," "Sample Correlation – Gene Expression," and "Sample Correlation – Methylation" illustrate the pairwise correlation between samples based on three different molecular data types. Each heatmap uses hierarchical clustering to

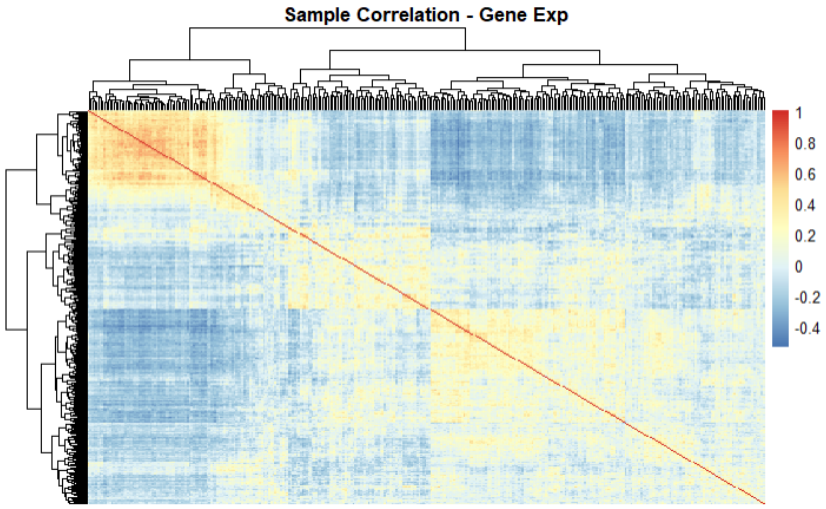

Figure 8: Sample Correlation - Gene Experession

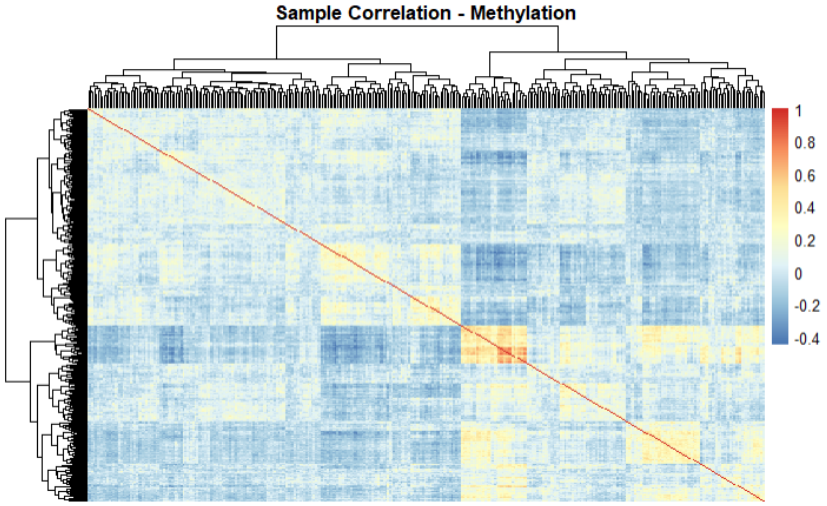

Figure 9: Sample Correlation - Methylation

reveal patterns of similarity and dissimilarity between samples, with a color scale ranging from blue (negative correlation) to red (positive correlation). The strong red diagonal in each heatmap represents perfect self-correlation, while the varying intensity of off-diagonal cells highlights the degree of correlation between different samples. Distinct clusters of red along the diagonals suggest the presence of biologically meaningful subgroups, potentially indicating cancer subtypes or underlying molecular phenotypes. Differences in clustering patterns across the three data types provide complementary insights into the molecular landscape of the samples. For instance, while gene expression and miRNA profiles may reflect active regulatory processes, methylation patterns might point to more stable epigenetic modifications. These visualizations help identify consistent sample groupings across data types and guide further integrative analyses such as multi-omics clustering or subtype characterization.

All the codes are presented in the Appendix. The summary output from the model run on the breast cancer dataset using feature selection (`VS_method = 1`) and a maximum of six clusters ($K = 6$) provides meaningful insight into the structure of the data. Among 348 individuals, the algorithm determined an optimal solution with five distinct clusters: Cluster 2 with 137 members, Cluster 6 with 81, Cluster 1 with 65, Cluster 3 with 38, and Cluster 5 with 27. This stratification suggests the presence of multiple biologically heterogeneous subgroups within the breast cancer cohort.

Feature selection outcomes indicate the model successfully identified informative variables across all three data views. Specifically, it selected 488 out of 645 features from gene expression data (View 1), 437 out of 574 from DNA methylation data (View 2), and 193 out of 423 from miRNA expression data (View 3), each surpassing the posterior inclusion probability threshold of 0.5. These results highlight the varying degrees of contribution from each omics layer in distinguishing subtypes. The relatively high number of selected features across all views suggests strong multivariate signals in the data, while the differences in selection ratios may reflect underlying biological mechanisms or measurement variability across platforms. The `piplot()` function can further visualize the distribution of posterior inclusion probabilities and assist in refining thresholds where needed.

The final ELBO value of $-57{,}406{,}131.13$ after 100 iterations indicates the model's convergence and optimization of the variational Bayes objective. Heatmaps for each data view can offer visual interpretation of the molecular profiles associated with each cluster. For example, gene expression heatmaps may reveal distinct transcriptional programs, while methylation heatmaps can illustrate epigenetic divergence. Although not yet analyzed, Kaplan–Meier survival plots could provide critical information on the clinical relevance of these clusters, offering insights into potential prognosis or therapeutic response.

5 Conclusion

In this project, we thoroughly examine the algorithm presented in the `iClusterVB` paper and look at each aspect of the method to gain a complete understanding of how it works. We explored key components such as the finite mixture model, feature selection, prior distributions, and the application of variational inference to estimate model parameters. To solidify our grasp of the approach, we implemented the method on simulated data, replicating the experimental setup outlined in the paper. This helped us to not only deepen our understanding of the theoretical aspects but also observe the algorithm in action.

The results demonstrated that `iClusterVB` is a robust tool for clustering and feature selection, particularly when dealing with datasets that have many variables of different types. This exercise highlighted the method's potential applications in real-world genomics challenges, such as identifying disease subtypes and discovering biomarkers.

Overall, this project provided us with both theoretical insights and hands-on experience with a cutting-edge approach in modern data science.

The findings from our work can be summarized as follows:

- **Summary of Findings:** The `iClusterVB` method excels in clustering datasets with diverse feature types, including continuous, binary, and count data. It effectively identifies underlying patterns and groups similar samples by selecting the most significant features.

- **Practical Implications:** This method is particularly valuable in genomics and biomedical research, especially in multi-omics studies that integrate data from various sources, such as gene expression, mutations, and copy number variations. It is also instrumental in identifying biomarkers and classifying patients into meaningful subgroups, enhancing treatment strategies and our understanding of diseases.

- **Future Directions:** Moving forward, there are opportunities to enhance this method by incorporating supervised learning with labeled data, such as cancer subtypes. Additionally, applying the algorithm to real-world cancer datasets will help assess its performance and potentially lead to new scientific discoveries.

6 References

References

[1] A. Alnajjar and Z. Lu, *iClusterVB: An R package for fast integrative clustering and feature selection with applications to multi-view cancer data*, Preprint, Elsevier, 2024. Available at: `https://doi.org/10.48550/arXiv.2403.14121`.

[2] Z. Lu, A. Alnajjar, and H. Bian, *Fast integrative clustering and feature selection for high-dimensional data: A variational Bayes approach*, *Journal of Computational and Graphical Statistics*, under review, 2024.

[3] Z. Lu, *A fast integrative clustering and feature selection approach for high-dimensional data*, Presentation, Department of Public Health Sciences & Department of Mathematics and Statistics, Queen's University, 2024.

[4] L. Buitinck, G. Louppe, and others, *API design for machine learning software: Experiences from the scikit-learn project*, In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

7 Appendix

7.1 Tables

Table 9: Normality test for Gaussian distribution (first 1000 columns in the simulation data)

Index	Variable	Skewness	Kurtosis	Normality p-value
0	gauss_2_396	-0.1079	-0.0766	0.7817
1	gauss_1_225	0.0277	-0.1564	0.9313
2	gauss_2_18	-0.0052	-1.5656	0.0000
3	gauss_1_90	-0.1858	0.3446	0.2363
4	gauss_1_375	0.0521	-0.0305	0.9374
5	gauss_2_217	-0.0205	-0.1222	0.9708
6	gauss_1_490	0.0701	-0.0306	0.8950
7	gauss_1_260	-0.0380	0.0563	0.8946
8	gauss_1_124	0.1376	-0.2938	0.4495
9	gauss_1_400	-0.1274	-0.3238	0.4176
10	gauss_1_167	0.1571	-0.3698	0.2736
11	gauss_1_18	-0.0073	-1.5921	0.0000
12	gauss_1_318	0.0031	0.1366	0.8125
13	gauss_1_220	0.1701	-0.4797	0.1043
14	gauss_1_308	-0.0471	-0.3887	0.3883
15	gauss_1_385	0.0960	-0.1883	0.7422
16	gauss_2_210	0.0282	0.3164	0.5181
17	gauss_2_354	-0.0808	0.2698	0.5219
18	gauss_1_369	0.0930	0.0298	0.7920
19	gauss_1_147	-0.2302	0.1885	0.2441
20	gauss_1_50	-0.0317	-1.6111	0.0000
21	gauss_1_405	-0.0403	0.0832	0.8590
22	gauss_1_117	0.0661	-0.3219	0.5403
23	gauss_2_179	-0.0018	-0.3278	0.5762
24	gauss_2_249	0.3049	0.6479	0.0258
25	gauss_1_68	-0.1215	-0.1024	0.7268
26	gauss_1_299	0.2415	-0.1434	0.2882
27	gauss_1_118	-0.2193	0.0210	0.3537
28	gauss_2_48	0.0079	-1.5732	0.0000
29	gauss_2_484	0.1617	-0.2456	0.4565

Table 10: Poisson test for Poisson distribution (1500 to 2000 columns in the simulation data)

Index	Variable	Statistic	Poisson p-value
0	poisson_1_347	8.1418	0.4197
1	poisson_1_130	3.3996	0.8457
2	poisson_1_278	2.0546	0.9568
3	poisson_1_31	1896309619.6291	0.0000
4	poisson_1_233	2.1521	0.9509
5	poisson_1_405	7.6532	0.3642
6	poisson_1_343	4.8117	0.6829
7	poisson_1_458	4.6286	0.5923
8	poisson_1_336	9.0090	0.3415
9	poisson_1_222	7.1677	0.4116
10	poisson_1_440	6.7327	0.3463
11	poisson_1_357	7.3086	0.3975
12	poisson_1_341	9.9632	0.1907
13	poisson_1_113	5.7099	0.4565
14	poisson_1_420	1.8409	0.9337
15	poisson_1_447	12.7249	0.0476
16	poisson_1_248	7.0096	0.5356
17	poisson_1_417	7.0637	0.2159
18	poisson_1_194	2.7987	0.8337
19	poisson_1_9	6684971.6159	0.0000
20	poisson_1_61	2.9649	0.8132
21	poisson_1_22	952499618.9309	0.0000
22	poisson_1_407	0.9116	0.9887
23	poisson_1_282	41.5326	0.0000
24	poisson_1_129	8.1200	0.4218
25	poisson_1_421	7.0042	0.4284
26	poisson_1_204	24.9668	0.0008
27	poisson_1_427	3.0758	0.8779
28	poisson_1_258	10.4206	0.1080
29	poisson_1_383	5.0726	0.6511

Table 11: Bernoulli test for Bernoulli distribution (1000 to 1500 columns in the simulation data)

Index	**Variable**	**Chi2 Statistic**	**p-value**	**Estimated p**	**# 0s**	**# 1s**
0	multinomial_1_353	0.7407	0.3894	0.1167	212	28
1	multinomial_1_345	1.6667	0.1967	0.0750	222	18
2	multinomial_1_196	1.6667	0.1967	0.0750	222	18
3	multinomial_1_103	1.1574	0.2820	0.1208	211	29
4	multinomial_1_83	1.1574	0.2820	0.0792	221	19
5	multinomial_1_499	1.6667	0.1967	0.0750	222	18
6	multinomial_1_22	106.6667	0.0000	0.3000	168	72
7	multinomial_1_138	0.4167	0.5186	0.0875	219	21
8	multinomial_1_360	4.6296	0.0314	0.1417	206	34
9	multinomial_1_72	0.7407	0.3894	0.1167	212	28
10	multinomial_1_486	0.0000	1.0000	0.1000	216	24
11	multinomial_1_101	2.9630	0.0852	0.0667	224	16
12	multinomial_1_90	0.0463	0.8296	0.0958	217	23
13	multinomial_1_42	161.1574	0.0000	0.3458	157	83
14	multinomial_1_271	0.0463	0.8296	0.0958	217	23
15	multinomial_1_252	0.0000	1.0000	0.1000	216	24
16	multinomial_1_110	0.7407	0.3894	0.1167	212	28
17	multinomial_1_415	0.7407	0.3894	0.1167	212	28
18	multinomial_1_122	0.7407	0.3894	0.0833	220	20
19	multinomial_1_230	0.4167	0.5186	0.0875	219	21
20	multinomial_1_403	1.6667	0.1967	0.0750	222	18
21	multinomial_1_88	0.4167	0.5186	0.1125	213	27
22	multinomial_1_126	7.8241	0.0052	0.1542	203	37
23	multinomial_1_130	1.1574	0.2820	0.0792	221	19
24	multinomial_1_439	0.4167	0.5186	0.1125	213	27
25	multinomial_1_16	89.6296	0.0000	0.2833	172	68
26	multinomial_1_340	3.7500	0.0528	0.1375	207	33
27	multinomial_1_293	0.0000	1.0000	0.1000	216	24
28	multinomial_1_64	0.0000	1.0000	0.1000	216	24
29	multinomial_1_82	0.4167	0.5186	0.0875	219	21

Table 12: Normality test for all variables in the breast cancer dataset

Index	**Variable**	**Skewness**	**Kurtosis**	**Normality p-value**
0	id	6.9517	48.7647	0.0000
1	radius_mean	-0.4930	0.2500	0.0006
2	texture_mean	0.3480	0.1690	0.0207
3	perimeter_mean	-0.5277	0.3695	0.0002
4	area_mean	-0.5650	0.3883	0.0001
5	smoothness_mean	0.6002	1.5932	0.0000
6	compactness_mean	1.0849	1.7566	0.0000
7	concavity_mean	2.9089	15.0911	0.0000
8	concave points_mean	0.8717	0.8859	0.0000
9	symmetry_mean	0.5683	1.0837	0.0000
10	fractal_dimension_mean	1.6090	4.2304	0.0000
11	radius_se	1.1020	2.0694	0.0000
12	texture_se	0.5976	0.5768	0.0000
13	perimeter_se	0.5015	-0.0299	0.0009
14	area_se	0.1348	0.2530	0.3185
15	smoothness_se	1.4897	2.9455	0.0000
16	compactness_se	2.1325	5.4932	0.0000
17	concavity_se	5.4665	45.6831	0.0000
18	concave points_se	2.0728	9.9365	0.0000
19	symmetry_se	1.3380	3.0752	0.0000
20	fractal_dimension_se	4.2709	26.7974	0.0000
21	radius_worst	-0.4131	0.0001	0.0073
22	texture_worst	0.1375	-0.2272	0.3988
23	perimeter_worst	-0.4131	0.0143	0.0072
24	area_worst	-0.4709	0.0885	0.0017
25	smoothness_worst	0.2805	0.2833	0.0477
26	compactness_worst	0.8333	0.8487	0.0000
27	concavity_worst	1.6051	5.2030	0.0000
28	concave points_worst	0.0221	-0.2058	0.7610
29	symmetry_worst	0.1265	0.1762	0.4333
30	fractal_dimension_worst	1.3515	2.8535	0.0000

7.2 R code-1

```r
# Load the iClusterVB package
library(iClusterVB)
# Load the cowplot library to combine plots
library(cowplot)

# Part 1: simulation data
# Load the built-in simulated data
get(data("sim_data"))

# Create a list of data views from the simulated dataset
list_sim_data <- list(
  gauss_1 = sim_data$continuous1_data,
  gauss_2 = sim_data$continuous2_data,
  multinomial_1 = sim_data$binary_data,
  poisson_1 = sim_data$count_data
)

# Re-code 0's to 2's for the multinomial data (iClusterVB requires
   non-zero values)
list_sim_data$multinomial_1[list_sim_data$multinomial_1 == 0] <- 2

# Optional: check first few rows of each dataset
head(list_sim_data$gauss_1[, 1:6])
head(list_sim_data$gauss_2[, 1:6])
head(list_sim_data$multinomial_1[, 1:6])
head(list_sim_data$poisson_1[, 1:6])

# Specify the distribution for each view
dist_sim_data <- c("gaussian", "gaussian", "multinomial", "poisson")

# Run the iClusterVB model on the simulated data
set.seed(123)
fit_sim_data <- iClusterVB(
  mydata = list_sim_data,
  dist = dist_sim_data,
  K = 8,  # max clusters to allow algorithm to reduce
  initial_method = "VarSelLCM",
```

```r
  VS_method = 1,  # enable feature selection
  max_iter = 100
)

# Summarize the fitted model
summary(fit_sim_data, rho = 0.5)

# Compare estimated clusters with ground truth
table(fit_sim_data$cluster, sim_data$cluster_true)

# plot piplot
piplot(fit_sim_data , nrow = 2, ncol = 2, align = "hv␣")

# Generate cluster-specific heatmaps from iClusterVB results
hmaps_sim <- chmap(
  fit_sim_data,
  rho = 0,
  cols = c(
    "#000000",  # black
    "#F8766D",  # reddish-pink
    "#00BA38",  # green
    "#619CFF"   # blue
  ),
  scale = "none"  # no scaling applied to preserve original feature scale
)

# Arrange the generated heatmaps in a 2x2 grid layout
plot_grid(
  plotlist = hmaps_sim,
  ncol = 2,
  nrow = 2,
  labels = c("(A)", "(B)", "(C)", "(D)")
)
```

Listing 1: R code example

7.3 R code - Breast Cancer data set one

```r

# implement on Breast Cancer data set
normal_brac <- read.csv("../data/breast_cancer_b_normal_variables.csv")
pois_brac <- read.csv("../data/breast_cancer_b_poisson_variables.csv")

# Convert to matrices
normal_brac_matrix <- as.matrix(normal_brac)
pois_brac_matrix <- as.matrix(pois_brac)

# create the list of real data
real_data <- list(normal_brac = normal_brac_matrix,
                  pois_brac = pois_brac_matrix)

# Specify the distribution for each view
dist_real_data <- c("gaussian", "poisson")

fit_real_data <- iClusterVB(
  mydata = real_data,
  dist = dist_real_data,
  K = 6,  # max clusters to allow algorithm to reduce
  initial_method = "VarSelLCM",
  VS_method = 1,  # enable feature selection
  max_iter = 500
)

# report the result

# Summarize the fitted model
summary(fit_real_data, rho = 0.5)

# Compare estimated clusters with ground truth
table(fit_sim_data$cluster, sim_data$cluster_true)

# plot piplot
piplot(fit_real_data , nrow = 2, ncol = 2, align = "hv␣")
```

```r

# Generate cluster-specific heatmaps from iClusterVB results
hmaps_sim <- chmap(
  fit_real_data,
  rho = 0,
  cols = c(
    "#000000",  # black
    "#F8766D",  # reddish-pink
    "#00BA38",  # green
    "#619CFF",   # blue
    "yellow",
    "orange"
  ),
  scale = "none"  # no scaling applied to preserve original feature scale
)

# Arrange the generated heatmaps in a 2x2 grid layout
plot_grid(
  plotlist = hmaps_sim,
  ncol = 2,
  nrow = 2,
  labels = c("(A)", "(B)", "(C)", "(D)")
)
```

Listing 2: R code example

7.4 R code-BRCA TCG

```r
```{r}
# Load and transpose breast cancer data from CSV files
gene_exp <- read.csv(file = "C:...Project2/Breast-TCGA/Breast-TCGA/
   context1_GE.csv", header = TRUE, row.names = 1)
gene_exp <- t(gene_exp)

methy_exp <- read.csv(file = "C:...Project2/Breast-TCGA/Breast-TCGA/
   context2_Meth.csv", header = TRUE, row.names = 1)
methy_exp <- t(methy_exp)

mirna_exp <- read.csv(file = "C:...Project2/Breast-TCGA/Breast-TCGA/
   context3_miRNA.csv", header = TRUE, row.names = 1)
mirna_exp <- t(mirna_exp)

# Load clinical/survival data
clinical <- read.csv(file = "C:/Users/hadis/OneDrive/Documents/
   seminar/Project2/Breast-TCGA/Breast-TCGA/Table1Nature.csv",
   header = TRUE)


# Extract  PatientID
id <- sub("TCGA-\\w+-(\\w+)", "\\1", clinical$PatientID)
clinical$id <- id

# Scale the data
gene_exp_scale <- apply(gene_exp, 2, scale)
methy_exp_scale <- apply(methy_exp, 2, scale)
mirna_exp_scale <- apply(mirna_exp, 2, scale)

# Combine into a list for iClusterVB input
list_brca_data <- list(
  gene_exp_scale,
  methy_exp_scale,
  mirna_exp_scale
)

# Specify data distribution for each view
```
```

```r
dist_brca_data <- c("gaussian", "gaussian", "gaussian")

library("iClusterVB")

set.seed(20204)

fit_brca <- iClusterVB(
  mydata = list_brca_data,
  dist = dist_brca_data,
  K = 6,
  initial_method = "VarSelLCM",
  initial_vs_prob = 0.1,        # Prior probability of variable selection
  max_iter = 100,               # Max iterations for convergence
  VS_method = 1,
  per = 100
)

summary(fit_brca)

library(ggplot2)

var_gene <- apply(gene_exp_scale, 2, var)
var_methy <- apply(methy_exp_scale, 2, var)
var_mirna <- apply(mirna_exp_scale, 2, var)

df_var <- data.frame(
  variance = c(var_gene, var_methy, var_mirna),
  type = c(rep("Gene", length(var_gene)), rep("Methy", length(var_methy)), rep("miRNA", length(var_mirna)))
)

ggplot(df_var, aes(x = variance, fill = type)) +
  geom_histogram(bins = 50, alpha = 0.7, position = "identity") +
  scale_x_log10() +
  theme_minimal() +
  ggtitle("Feature␣Variance␣Distribution␣Across␣Omics")

```

```r
library(pheatmap)

pheatmap(cor(t(gene_exp_scale)), show_rownames = FALSE, show_
   colnames = FALSE, main = "Sample Correlation - Gene Exp")
pheatmap(cor(t(methy_exp_scale)), show_rownames = FALSE, show_
   colnames = FALSE, main = "Sample Correlation - Methylation")
pheatmap(cor(t(mirna_exp_scale)), show_rownames = FALSE, show_
   colnames = FALSE, main = "Sample Correlation - miRNA")


pca_plot <- function(data, title) {
  pca <- prcomp(data, center = TRUE, scale. = TRUE)
  pca_df <- data.frame(PC1 = pca$x[,1], PC2 = pca$x[,2])

  ggplot(pca_df, aes(x = PC1, y = PC2)) +
    geom_point(alpha = 0.6) +
    ggtitle(title) +
    theme_minimal()
}

pca_plot(gene_exp_scale, "PCA - Gene Expression")
pca_plot(methy_exp_scale, "PCA - Methylation")
pca_plot(mirna_exp_scale, "PCA - miRNA Expression")
```

Listing 3: R code example