# Nonparametric clustering of RNA-sequencing data

Gabriel Lozano [*] Nadia Atallah[†] Michael Levine[‡]

May 28, 2023

**Abstract**

Identification of clusters of co-expressed genes in transcriptomic data is a difficult task. Most algorithms used for this purpose can be classified into two broad categories: distance-based or model-based approaches. Distance-based approaches typically utilize a distance function between pairs of data objects and group similar objects together into clusters. Model-based approaches are based on using the mixture-modeling framework. Compared to distance-based approaches, model-based approaches offer better interpretability because each cluster can be explicitly characterized in terms of the proposed model. However, these models present a particular difficulty in identifying a correct multivariate distribution that a mixture can be based upon. In this manuscript, we review some of the approaches used to select a distribution for the needed mixture model first. Then, we propose avoiding this problem altogether by using a nonparametric MSL (Maximum Smoothed Likelihood) algorithm. This algorithm was proposed earlier in statistical literature but has not been, to the best of our knowledge, applied to transcriptomics data. The salient feature of this approach is that it avoids explicit specification of distributions of individual biological samples altogether, thus making the task of a practitioner easier. We perform both a simulation study and an application of the proposed algorithm to two different real datasets. When used on a real dataset, the algorithm produces a large number of biologically meaningful clusters and performs at least as well as several other mixture-based algorithms commonly used for RNA-seq data clustering. Our results also show that this algorithm is capable of uncovering clustering solutions that may go unnoticed by several other model-based clustering algorithms. Our code is publicly available in Github at `https://github.com/Matematikoi/non_parametric_clustering`

[*]Universidad Nacional de Colombia, golozanop@unal.edu.co

[†]Department of Comparative Pathobiology, Purdue Univesity, natallah@purdue.edu

[‡]Department of Statistics, Purdue University, mlevins@purdue.edu

# 1 Introduction

Increasingly complex studies of transcriptome dynamics can be carried out now using high-throughput sequencing of reverse-transcribed RNA molecules. Such a procedure is typically called RNA-sequencing (RNA-seq). Studying RNA-seq data helps researchers gain a deeper understanding of how changes in transcriptional activity reflect various cell types and contribute to phenotypic differences. One of the ways to gain new insights from RNA-seq data is to identify groups (clusters) of co-expressed genes. This can help researchers target genes involved in similar biological processes, thus helping in the design of new pharmaceutical drugs, or finding genes that are candidates for co-regulation. Identification of clusters of co-expressed genes also helps characterize biological functions for orphan genes.

By now, a number of algorithms has been proposed for clustering RNA-seq data. Note that these data have a number of characteristics that makes modeling them rather difficult. First of all, they tend to be highly skewed and have a large dynamic range. Second, they typically demonstrate positive correlation between the gene length and read counts. Third, these data are almost always overdispersed (i.e. their variance is larger than their mean).

Most of the model-based clustering methods used for RNA-seq data typically view each cluster as represented by a distinct distribution, while the entire dataset is modeled as a finite mixture of these distributions. One of the advantages of this approach is that it allows a researcher to assess the appropriate number of clusters, the distance between clusters, and test hypotheses about these quantities. In the existing literature, most of these models tend to be parametric; in other words, it is assumed that distributions of individual biological samples are modeled as belonging to a particular distribution. The choice of distribution ranges from Poisson [1] to negative binomial [2] to complicated distributions built specifically for the given task, such as multivariate Poisson-lognormal [3, 4]. The choice that has to be made here is a rather difficult one as there are few ready-made goodness of fit tests for most multivariate discrete distributions. Although parametric approaches enjoy, as a rule, lower computational costs and have a benefit of rather simple interpretation, it may be beneficial to look for a possible alternative. Such an alternative can be provided by the use of nonparametric finite mixtures for biological clustering. In particular, nonparametric modeling makes very few assumptions about the nature of the analyzed data.

More specifically, we suggest viewing the distributions used to model components of a mixtures as "just" probability density functions that do not belong to any such family. The nonparametric approach to multivariate mixture modeling and clustering has been studied in statistics

for some time; several methods have been proposed to fit such models and establish the structure of relevant clusters [5, 6, 7]. However, their use in bioinformatics in general has been very limited (see e.g. [8]); to the best of our knowledge, they have not been used to cluster transcriptomic data at all.

In this work, we suggest the use of an algorithm that can fit a general multivariate nonparametric mixture model with conditionally independent marginals in the RNA-seq data context. This method is a so-called npMSL (the nonparametric Maximum Smoothed Likelihood) method that was originally proposed in [6]. The corresponding algorithm is an MM (Maximization-Minorization) algorithm that possesses the monotonicity property, similarly to the EM algorithm, and is guaranteed to converge.

The remainder of this article is organized as follows. In the Methods section, we describe the model used in detail, introduce the necessary notation, and define the algorithm that is to be used to perform the clustering task. Here, we also describe the model selection procedure that we use to select an appropriate number of clusters. Next follows the Simulation section that uses a synthetic dataset to analyze the proposed method. In the Real Data Analysis section, we describe in detail a human prostate cancer dataset that we use to illustrate our approach. The results are illustrated using both a simple visualization method and the results of a GO functional enrichment analysis. This study suggests that the proposed algorithm may be able of uncovering practically important clustering solutions that may go unnoticed by other mixture model based clustering algorithms. Some concluding remarks are provided in the Discussion section. Finally, the performance of the proposed algorithm is studied using an additional mouse tissue dataset in the Appendix.

# 2 Methods

Let $Y_{ij}$ be the random variable corresponding to the digital gene expression measure for a biological entity $i$ ($i = 1, \ldots, n$) of condition $j$ ($j = 1, \ldots, d$). We denote the corresponding observed value $y_{ij}$. This setting implies that the data $\mathbf{y}$ is the $n \times d$ matrix of the digital gene expression for all observations and variables. Also, $\mathbf{y}_i$ is the $d-$ dimensional vector of digital gene expression for all variables of observation $i$.

## 2.1 Nonparametric mixture model with conditionally independent measurements

Historically, it has been common to use parametric mixture models to cluster RNA-seq data (see e.g.[2, 1, 9]). To the best of our knowledge, nonparametric approach to model-based clustering of RNA-seq data has not been tried before. At the same time, it has been used in many other statistical application areas e.g. developmental psychology, hydrology, and others [7, 10]. This approach assumes that the density functions of clusters do not come from a particular parametric family (e.g. Gaussian, Poisson etc.). More specifically, the data are assumed to come from $m$ distinct clusters with the density of the $k$th cluster, $k = 1, \ldots, m$ being $f_k$. It is further assumed that each one of these densities $f_k$ is equal with probability 1 to the product of its marginal densities:

$$f_k(\mathbf{y}_i) = \prod_{j=1}^{d} f_{kj}(y_{ij}) \tag{1}$$

with $\mathbf{y}_i = (y_{i1}, \ldots, y_{id})'$. Taking a fully nonparametric approach with regard to the $f_{kj}$, we may therefore express the density function of any observation $\mathbf{Y}_i$ according to a nonparametric finite mixture model as

$$\mathbf{Y}_i \ \sim \ g_{\boldsymbol{\theta}}(\mathbf{y}_i) \ = \ \sum_{k=1}^{m} \pi_k \prod_{j=1}^{d} f_{kj}(y_{ij}), \tag{2}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$ must satisfy

$$\sum_{k=1}^{m} \pi_k = 1 \quad \text{and each } \pi_k \geq 0. \tag{3}$$

Here, we assume $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{id})'$ and we let $\boldsymbol{\theta}$ denote the vector of parameters to be estimated, including the mixing proportions $\pi_1, \ldots, \pi_m$ and the univariate densities $f_{kj}$. For convenience, we will also use the notation $\mathbf{f}$ for a vector of all the marginal densities $\{f_{kj}\}$, $j = 1, \ldots, d$, $k = 1, \ldots, m$. Furthermore, throughout this article, $k$ and $j$ always denote the component and coordinate indices, respectively; thus, $1 \leq j \leq d$ and $1 \leq k \leq m$. Thus, the overall population of the sample size $n$ is distributed according to

$$f(\mathbf{y}_i; m, \boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{m} \pi_k \prod_{j=1}^{d} f_{kj}(y_{ij}). \tag{4}$$

The conditional independence may seem rather limiting at first sight. However, it may be thought of as a simplification of the commonly used repeated measures random effects model. In such a model,

one usually assumes that the multivariate observations on an individual are independent, conditional on the identity of the individual in question. Here, the individual-level effects are replaced by the entity (gene)-level effects. Note also that the conditional independence assumption has been used on a number of occasions when modeling RNA-seq data using parametric mixtures e.g. [1, 2]. On the other hand, no individual marginal density $f_{jk}$ is assumed to have come from a family of densities indexed by a finite-dimensional parameter vector, such as Gaussian, Student, etc. Such a nonparametric approach represents a substantial generalization compared to approaches typically used in bioinformatics literature.

## 2.2 Inference

We are going to start with some needed notation. First, let $Z_{ik} \in \{0, 1\}$ be a Bernoulli random variable indicating that an individual $i$ comes from the component $k$. Because each individual comes from exactly one component, it implies that $\sum_{k=1}^{m} Z_{ik} = 1$. Therefore, the complete data would be the set of all $(\mathbf{x}_i, \mathbf{Z}_i)$, where $1 \leq i \leq n$. This suggests that the entire data can be viewed as consisting of observable and unobservable parts. Therefore, an EM-type algorithm seems to be an appropriate option for estimating of parameters of the model (2).

[6] introduced an algorithm that minimizes a smoothed loglikelihood function of the data produced by the model (2). This algorithm has a provable monotonicity property. We only give a brief description of this algorithm; for detailed discussion of its monotonicity property see [6]. For the purpose of this discussion, some additional notation is needed.

Let $K(\cdot)$ be a kernel density function on the real line. With a slight abuse of notation, let us define the product kernel function in the $d$-dimensional space as $K(\mathbf{u}) = \prod_{j=1}^{d} K(u_j)$ and its rescaled version $K_h(\mathbf{u}) = h^{-d} \prod_{j=1}^{d} K(h^{-1}u_j)$ for a positive parameter $h$ that is commonly called the bandwidth. Furthermore, we smooth a function $f$ using the following smoothing operation $\mathcal{S}f(\mathbf{y}) = \int K_h(\mathbf{y}-\mathbf{u})f(\mathbf{u})\,d\mathbf{u}$. The same smoothing operation can be applied to an $m$-dimensional vector of functions by defining $\mathcal{S}\mathbf{f} = (\mathcal{S}f_1, \ldots, \mathcal{S}f_m)'$. We also define a nonlinear smoothing operation $\mathcal{N}$ as

$$\mathcal{N}f(\mathbf{y}) \;=\; \exp\{(\mathcal{S}\log f)(\mathbf{y})\} \;=\; \exp\int K_h(\mathbf{y}-\mathbf{u})\log f(\mathbf{u})\,d\mathbf{u}.$$

To simplify notation, we introduce the finite mixture operator $\mathcal{M}_{\boldsymbol{\pi}}\mathbf{f}(\mathbf{y}) := \sum_{k=1}^{m} \pi_k f_k(\mathbf{y})$, whence we obtain $\mathcal{M}_{\boldsymbol{\pi}}\mathbf{f}(\mathbf{y}) = g_{\boldsymbol{\theta}}(\mathbf{y})$. Also, we denote $\mathcal{M}_{\boldsymbol{\pi}}\mathcal{N}\mathbf{f}(\mathbf{y}) := \sum_{k=1}^{m} \pi_k \mathcal{N}f_k(\mathbf{y})$. With this notation in mind, we define the following algorithm. Given initial values $(\mathbf{f}^0, \boldsymbol{\pi}^0)$, iterate the following three steps for $t = 0, 1, \ldots$ :

- **E-step:** Define, for each $i$ and $k$,

$$w_{ik}^t = \frac{\pi_k^t \mathcal{N} f_k^t(\mathbf{y}_i)}{\mathcal{M}_{\boldsymbol{\pi}^t} \mathcal{N} \mathbf{f}^t(\mathbf{y}_i)} = \frac{\pi_k^t \mathcal{N} f_k^t(\mathbf{y}_i)}{\sum_{a=1}^m \pi_a^t \mathcal{N} f_a^t(\mathbf{y}_i)}. \tag{5}$$

- **M-step, part 1:** Set

$$\pi_k^{t+1} = \frac{1}{n} \sum_{i=1}^n w_{ik}^t \tag{6}$$

  for $k = 1, \ldots, m$.

- **M-step, part 2:** For each $j$ and $k$, let

$$\begin{aligned} f_{kj}^{t+1}(u) &= \frac{\sum_{i=1}^n w_{ik}^t K_h(u - y_{ij})}{\sum_{i=1}^n w_{ik}^t} \\ &= \frac{1}{nh\pi_k^{t+1}} \sum_{i=1}^n w_{ik}^t K\left(\frac{u - y_{ij}}{h}\right). \end{aligned} \tag{7}$$

Let us define the following functional of $\boldsymbol{\theta}$ (and, implicitly, $g$):

$$\ell(\boldsymbol{\theta}) = \int g(\mathbf{y}) \log \frac{g(\mathbf{y})}{[\mathcal{M}_{\boldsymbol{\pi}} \mathcal{N} \mathbf{f}](\mathbf{y})} \, d\mathbf{y}. \tag{8}$$

This functional represents conceptually a smoothed negative log-likelihood. Then, [6] shows that the value of this functional decreases at each step of the introduced algorithm. This algorithm will be referred to as npMSL (nonparametric Maximum Smoothed Likelihood) algorithm.

The npMSL algorithm can also be generalized to a model where there are blocks of coordinates that are identically distributed (in addition to being conditionally independent). If we let $b_j$ be the block index of the $j$th coordinate, where $1 \leq b_j \leq L$ and $L$ is the total number of such blocks, then the model (2) is modified as

$$g_{\boldsymbol{\theta}}(\mathbf{y}_i) \;=\; \sum_{k=1}^m \pi_k \prod_{j=1}^d f_{kb_j}(y_{ij}).$$

If all the blocks have the size 1, we are back to the original model (2). The nonlinear smoothing operator $\mathcal{N} f_k = \prod_{j=1}^d \mathcal{N} f_{kb_j}$ and definitions of $\mathcal{M}_{\boldsymbol{\pi}} \mathbf{f}$ and $\mathcal{M}_{\boldsymbol{\pi}} \mathcal{N} \mathbf{f}$ remain unchanged. The only element of the algorithm that actually needs an update is the density estimation step (7). For the $k$th component and block $l \in \{1, \ldots, L\}$, we now have

$$f_{kl}^{t+1}(u) = \frac{\sum_{j=1}^d \sum_{i=1}^n w_{ik}^t I_{b_j = l} K_h(u - y_{ij})}{\sum_{j=1}^d \sum_{i=1}^n w_{ik}^t I_{b_j = l}}$$

where $I_{b_j=l}$ is an indicator function of the event $b_j = l$. The block version of the npMSL algorithm is the one that we have used in our study. In other words, when applying the npMSL method, we assume that replicates of the same condition represent a block of identically distributed coordinates.

## 2.3 Bandwidth and kernel function selection issues

Of the two issues - kernel and bandwidth selection - mentioned in the heading above, the first is a simpler of the two. There seems to be a general consensus in the literature on the density estimation that the choice of the kernel function does not matter much, at least in terms of efficiency of resulting estimators; see e.g. [11] for a general discussion of this issue.

On the contrary, sensible choice of the bandwidth $h$ is a challenging problem. The form of the npMSL algorithm introduced here assumes that $h$ is the same for each component and coordinate, or block. It is straightforward to introduce component- and block-specific bandwidths $h_{jl}$. Note also that individual component densities are not observed in the mixture setting. This fact complicates selection of the bandwidth in the mixture setting.

From the practical viewpoint, we found out that selecting a constant bandwidth according to the so-called Silverman's rule of thumb [12], p. 48, works reasonably well. This method suggests choosing the bandwidth as

$$h = 0.9(nd)^{-1/5} \min\left\{SD, \frac{IQR}{1.34}\right\} \tag{9}$$

where $SD$ is the standard deviation, $IQR$ the interquartile range, and $nd$ is the size of the entire dataset. In this approach, the SD and IQR are the standard deviation and interquartile range of the $nd$ univariate observations that are available from $n$ $d$-dimensional observations.

Note that this is a rather crude method in the nonparametric mixture setting. It is possible that it may result in under- or oversmoothing. First, pooling all of the data implies that one treats all of the mixture components as though they are from the same distribution. This can lead to an inflation of the bandwidth, especially if the mixture components' centers are well-separated. This is true because, in such a case, the variability of the pooled dataset will be larger than that of the individual components. Similarly, if the vector coordinates are not identically distributed within each component/block, the bandwidth could be biased upward for the same reason. Yet operating in the opposite direction is the fact that the expression $nd$ in the formula defining the bandwidth above is an overestimate of the "true" sample

size. One can think of the "true" sample size from each component being approximately equal to $\lambda_k nd$.

The arguments above show first of all that it would be useful to know something about the mixture structure in order to select a bandwidth. This suggests an iterative procedure in which the value of $h$ is modified, and the algorithm reapplied, after the output from the algorithm is obtained. This, however, is going to result in the violation of monotonicity property of npMSL algorithm. For a more detailed discussion of this topic, see [13]. As the above suggests, a careful exploration of the bandwidth selection question is a research topic unto itself. Thus, to make our application of npMSL algorithm to the analysis of RNA-seq data simpler, we are only using the constant bandwidth value selected according to (9).

## 2.4 Selecting the number of clusters

The npMSL algorithm assumes that the number of clusters is known in advance. This is almost never true when working with biological data. Unlike the case of parametric mixture models, there are few if any practically feasible approaches to determining the number of clusters in a nonparametric mixture model with independent marginals. The first suggested partial solution of this problem appeared in [14] but it can only provide an estimate of a lower bound on the number of components, not of the number of components itself. A later idea of [15] proposes an estimator of the number of components in such a model that is consistent. However, this approach requires some additional assumptions on the joint distribution of the latent and observed variables which may not be true in transcriptomics applications. Moreover, this method requires practitioner to choose several auxiliary parameters while the theoretical guidance concerning their choice is completely absent.

Due to the above, we decided to pursue a different approach. Recall that a very wide variety of different distributions can be modeled using finite normal distributions [16]. The number of Gaussian components required to achieve this goal may be, however, quite large. This is especially likely to be so in the case of bulk RNA-seq data that possess a large dynamic range and tend to be strongly skewed. Moreover, these components may not at all have any practical interpretation and may not represent individual clusters present in the data. It is possible, however, that any cluster contained in the data may be fitted with a linear combination (that is, a finite mixture) of some subset of these Gaussian components as has been discussed in detail in [17]. Thus, this would be a situation where a cluster and finite mixture component need not be one and the same object.

The method proposed in [17] is as follows. As a first step, the

total number of Gaussian mixture components $K$ needs to be selected to fit the given multivariate density. One can also use all possible covariance matrix structures for these Gaussian components. Next, one would use a mclustBIC function from the mclust R package [18] to find an optimal finite Gaussian mixture for the given multivariate dataset where optimality is measured by BIC (Bayesian Information Criterion). From this setting, we proceed to establish a hierarchy of clusterings combining individual Gaussian components according to the entropy principle. The result is a unique soft clustering for each number of clusters from 1 to $K$. The optimal number of clusters is then selected in the data-driven way by using a piecewise linear regression fit for the rescaled entropy plot as described in [17]. For the detailed discussion of MCLUST family of mixture models see e.g. [19]. Since one of the rationales of using nonparametric mixtures is to avoid classical Gaussian assumptions on the components, we believe that the resulting non-Gaussian clustering solution may provide a possible benchmark for the number of clusters in our nonparametric model. Note that this method also produces its own clustering solution that we will use in the following sections of this manuscript for comparison purposes. We will refer to it as a combination clustering approach.

# 3 Simulation study

As a first step, we conduct a simulation study to compare our proposed method with three other model-based methods: a Poisson model-based method of [1], the transformation-based method of [9] and the combination method of [17]. To avoid the situation where a synthetic dataset has been made "to order" based on a particular distribution, we use a selection of datasets generated using negative binomial distributions that has been suggested in [2]. We will only give a brief description of the data generating scheme here as the detailed description can be found in [2].

First, we start with a brief general description of the model used to generate synthetic data. We assume that $N_{gij}$ is a count of reads mapped to gene $g$ for replicate $j$ of treatment $i$ for $g = 1, \ldots, G$, $i = 1, \ldots, I$ and $j = 1, \ldots, n_i$. Here, $G$ is the total number of genes considered, $I$ is the number of treatment groups, and $n_i$ is the number of replicates for $i$th treatment. Thus, the data consist of counts $N_{gij}$. It is assumed that the mean of $N_{gij}$ is $\lambda_{gij}$ such that

$$\log \lambda_{gij} = s_{gij} + \alpha_g + \beta_{gi} \tag{10}$$

where $\sum_{i=1}^{I} \beta_{gi} = 0$ while $s_{gij}$ is an offset term and $\alpha_g$ is a geometric mean expression level of $g$th gene across all treatments. Finally, $\beta_{gi}$ measures the gene expression level for $g$th gene in treatment $i$

relative to the overall mean expression. To model the overdispersion phenomenon, we model the variance of counts as

$$Var(N_{gij}) = \lambda_{gij} + \phi_g \lambda_{gij}^2 \tag{11}$$

where $\phi_g$ is a dispersion parameter.

Now we describe exactly how the model (10)-(11) is parameterized. The synthetic data are based on an experiment with three treatment groups and three replicates for each treatment group. It is assumed in advance that there are $K = 7$ different expression patterns in the data that correspond to clusters. Centers of these clusters are characterized by $\mu_k = \eta_\mu \delta_k$ where $\eta_\mu$ determines the magnitude of gene expression changes across treatments and $\delta_k$ is a three-dimensional vector that describes the pattern of changes for $k$th cluster, $k = 1, \ldots, K$. Each of the clusters has a distinct profile characterized by values of coordinates of $\delta_k$. For every $g$th gene in the study, $g = 1, \ldots, G$ a multivariate index $\mathbf{Z}_g^0 = \{Z_{gk}^0 : k = 1, \ldots, K\}$ is drawn from a multinomial distribution with equal probabilities. The choice $Z_{gk}^0 = 1$ means that $g$th gene belongs to $k$th cluster and $Z_{gk}^0 = 1$ otherwise. Given $Z_{gk}^0 = 1$, the gene expression profile is simulated as $\beta_g = \mu_k + \varepsilon_g$ where $\varepsilon_g = (\varepsilon_{g1}, \varepsilon_{g2}, \varepsilon_{g3})'$ is an added fluctuation around cluster center $\mu_k$ specifically for $g$th gene. Univariate random variables $\varepsilon_{gi}$ for $i = 1, 2, 3$ are sampled from $\eta_\mu \eta_\varepsilon X$ where $X \sim N(0, 0.2^2)$. The overall mean expression $\alpha_g$ is drawn from $\eta_\alpha Y$ where $Y \sim N(4, 1)$ and $\eta_\alpha$ controls the magnitude of average expression level. The dispersion parameter $\phi_g$ has been modeled as $\eta_\phi Z$ where $Z \sim \Gamma(0.75, 2)$ while changing values of $\eta_\phi$ allows for different levels of dispersion. Finally, the offset $s_{gij} \sim N(0, 1)$. Given these parameters, the gene expression count $N_{gij}$ is modeled as a negative binomial random variables with the mean $\exp(s_{gij} + \alpha_g + \beta_{gi})$ and dispersion $\phi_g$.

The performance of different clustering approaches is assessed by comparing resulting partitions with the original partitions of genes defined by $\mathbf{Z}^0 = \{\mathbf{Z}_g^0 : g = 1. \ldots, 10000\}$. As usual, a better performance is indicated by a stronger agreement between the two partitions. We used two specific statistics to perform these comparisons. The first one is the pairwise sensitivity which describes the proportion of pairs of genes that are clustered together among all pairs that had the same original assumption. The second one is the pairwise specificity which describes the proportion of pairs of genes that clustered to different groups among all pairs that had different original assignment. Both approaches had been used previously in clustering applications [20], [21], [2].

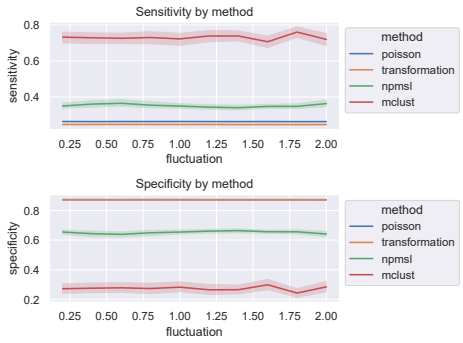
The performance of our method compared to the other three methods is illustrated under the two different assumptions. The first visualization (1a) shows what happens when the true number of clusters

$K = 7$ is known. The second one (1b) illustrates the situation where the number of clusters is assumed to be equal to 10. This is done since the true number of clusters is unknown in practice and we would like to illustrate the comparison between our proposed method and the two other methods under this misspecification as well. Note that the assumption about the number of clusters is only important for the npMSL, Poisson and transformation methods but not for the combination method. This is because the combination method determines the number of clusters automatically from the data. We let the fluctuation level $\eta_\varepsilon$ vary between 0.2 and 2 in steps of size 0.25 while keeping $\eta_\mu$, $\eta_\alpha$, and $\eta_\phi$ fixed at 1. For each combination of parameters stated above, we produce 100 datasets with that combination of parameters. For each resulting dataset, we compare the clustering result based on the use of npMSL and compare it with clustering results based on the Poisson method, the transformation method, and the combination method. While using the npMSL method, we used Silverman's rule of thumb as suggested in (9) to select the bandwidth and the Gaussian kernel for smoothing. Each comparison is performed using pairwise sensitivity and pairwise specificity. For both measures, we show the mean with a confidence band including 90% of results for all 100 generated datasets. For Poisson and transformation methods, these confidence bands are very narrow and cannot be seen well in plots. The plots in Figure (1a) - (1b) refer to the combination clustering method as the mclust method.

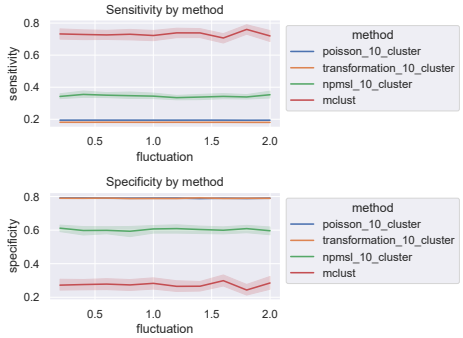
Under the correct specification of the number of clusters, our method exhibits sensitivity that is noticeably better than the sensitivity of the other two methods but inferior to the sensitivity of the combination method. In the plot describing the specificity, results for the Poisson-based method and the transformation-based method are very close to each other and cannot be separated visually. Here, npMSL method exhibits worse specificity than the other two methods but better then the specificity of the combination method. Note that the results are similar under the incorrect assumption on the number of clusters: once again, npMSL method beats the Poisson and transformation method in terms of sensitivity but performs worse than the combination method. The relationship reverses when it comes to specificity: npMSL specificity is consistently lower than that of either Poisson or transformation method but better than the specificity of the combination method.

# 4 Real data analysis

In the following, we illustrate the use of npMSL algorithm using two real RNA-seq datasets. Note that it is not possible to compare the co-expression results obtained using various clustering methods to a




(a) npMSL algorithm sensitivity and specificity compared to those of other methods under the correct cluster number assumption




(b) npMSL algorithm sensitivity and specificity compared to those of other methods under an incorrect cluster number assumption: $K = 10$

Figure 1: npMSL algorithm sensitivity and specificity in comparison to three other methods

"true" clustering of the data, as, in general, such a classification does not exist. In order to identify whether the co-expressed genes seem to be implicated in similar biological processes, we conduct functional enrichment analysis of gene ontology (GO) terms for the clusters identified by the suggested methods. The data that we are using are a mouse RNA-seq dataset consisting of lung, kidney, liver, and small intestine tissues [22] and a prostate cancer cell-line RNA-seq dataset [23]. The data are written in the matrix form where each row corresponds to a gene and each column to an experimental condition. The row names are the ENSEMBL gene names for each gene (ENSEMBL is a genome database project that is a scientific project of the European Bioinformatics Institute). Each row constitutes a digital gene expression of a particular gene across a set of cell lines (together with replicates) in our case. The goal is to cluster digital gene expression profiles in order to discover networks of co-expressed genes.

Unless explicitly stated to the contrary, both datasets are normalized first. The normalization procedure that we use is commonly called FPKM (Fragments Per Kilobase of transcript per Million mapped reads). Let us denote the result of this procedure $Y_i$ for the $i$th gene. Then, it is defined as

$$Y_i = \frac{X_i}{N * s_i} * 10^9$$

where $N$ is the total number of reads sequenced, $s_i$ is the length of gene i, and $X_i$ is the number of counts for the $i$th gene. This type of normalization is commonly used for visualization and clustering. It is necessary if we want to be able to perform within sample comparisons ("gene A is expressed higher or lower than gene B") because it is, effectively, a procedure that normalizes for gene length. Such a normalization is in order because, the longer the gene's length is, the more fragments ("reads") we sequence from that gene.

The data sets have also been filtered using a cutoff of 1.5 CPM (counts per million), in which gene read counts are divided by the sum of read counts for a given sample and multiplied by a million. The CPM of the $i$th gene, denoted as $CPM_i$ is defined as

$$CPM_i = \frac{X_i}{N} * 10^6$$

where $N$ is the total number of reads sequenced for a given sample and $X_i$ is the number of counts for the $i$th gene.

In what follows we give the results of the analysis based on the human prostate cancer cell line dataset. The analysis based on the second dataset is given in the Appendix.

### 4.1 Human prostate cancer cell line dataset

We begin with the description of the dataset. We have four cell lines used that consist of samples that are sensitive to chemotherapeutic agents (C4-2 cells and LNCaP cells) as well as those that are resistant (MR49F and C4-2B cells). More specifically, a line of C4-2 cells is a subline of LNCaP cells (described below). Second, C4-2B cells are enzalutamide-resistant cells derived from C4-2 cells. Third, LNCaP (Lymph Node Carcinoma of the Prostate) cell line is the line that has been established from a metastatic lesion of human prostatic adenocarcinoma. Finally, the MR49F cell line consists of enzalutamide-resistant cells derived from LNCaP cells. The resistant cell lines are of interest because these cells do not respond well to treatment. Also, each of the cell lines has three replicates. Filtering at the level of 1.5 CPM as described above results in a file containing sequencing data for $16,247$ genes.

It has been also noted earlier that the RNA-seq data often have a very large dynamic range and tend to be heavily skewed. This often represents a significant problem when modeling these data. Therefore, before modeling the data, we applied a logarithmic transform to it as a first step.

Note that it is unlikely that replicates in a cell line dataset are independent. If so, this dataset may violate a conditional independence model assumption that underlies the npMSL method. Due to this, an attempt to run our method on such a dataset may be viewed as a test of the algorithm's robustness to the violation of the conditional independence assumption. Indeed, we discovered that the algorithm tends not to converge if the dataset is used "as is" even when multiple choices of starting values are considered. Therefore, we decided to treat zeros as missing observations and use a simple imputation procedure, which turned out to be useful. More specifically, every zero was substituted with a number generated from a uniform distribution on an interval $[0, A]$ where $A$ was the smallest count observed in the entire dataset. We would like to note here that the practice of treating zeros as missing observations, although not very common in the analysis of RNA-seq data, is quite widespread when analyzing the scRNA-seq (Single Cell RNA-seq) data [24].

Using the method of selecting cluster numbers that we discussed earlier in section 2.4, we started with choosing the range from 1 to $K = 30$. The application of mclustBIC procedure suggests that, in a typical run, one needs about $29 - 30$ clusters with VVI or VVE covariance matrix structure. Here, VVI implies diagonal covariance matrices with different volumes and shapes for different components, with components parallel to coordinate axes. The VVE shape implies ellipsoidal covariance matrices with different volumes and shapes for

different components as well as the same orientation. The detailed discussion about all possible configurations of covariance matrices can be found in e.g. [19]. We found out that, in a typical run of the procedure, one ends up with 16-18 clusters as a possible suggestion. In what follows, we use 17 as a suggested cluster number.

For comparison purposes, we compare clustering results obtained using the npMSL method with results obtained using the three alternative methods: the Poisson method of [1] , the transformation-based method of [9] and the combination method of [17]. We believe that the Poisson method is an appropriate comparison benchmark since it, alongside our method, assumes conditional independence of marginals. Our method simply takes this assumption one step further and does not impose any specific distributional assumption on the marginal distributions. Note that the Poisson distribution only models the integer-valued data. Thus, when applying the Poisson method, we have used the raw data counts instead of the FPKM normalized dataset. We used the so-called slope heuristics method to select the optimal number of clusters in this case; our approach resulted in the choice of 18 clusters when running the slope heuristics approach over a range from 1 to 35 clusters.

Another method that we used for comparison purposes has been the transformation method of [9]. This method is based on the use of data transformations in conjunction with Gaussian mixture models. It also uses a penalized model selection criterion to select the number of clusters present in the data. Compared to the Poisson method, it allows for modeling of per-cluster correlation among biological samples. To analyze our dataset, we used this method with an arcsin transform. To select the number of clusters for this method we used the ICL (Integrated Complete Likelihood) criterion method [25]. Using this criterion over a possible range of 1 to 40 clusters, we found that 35 seemed to be the optimal number in this case.

The final comparison method that we use is a combination method of [17]. We believe that this method is a reasonable comparison benchmark for our method since it tries to account for clear non-normality of clusters commonly observed in transcriptomics data. The difference is that our method tries to fit clusters based completely on the data ("letting the data speak for themselves" in the common nonparametric parlance) while the method of [17] attempts to achieve this goal by modeling non-Gaussian clusters as mixtures of Gaussians. Note that this method also allows for per-cluster correlation among genes. Over a range of 1 to 30 possible clusters, this method suggests anywhere between 16 to 18 cluster solutions in a typical run of the procedure. We decided to use a 17 cluster solution in this case.

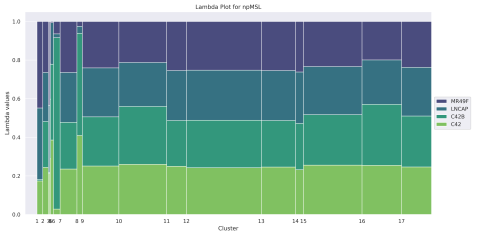
As is well known, visualizing results of a co-expression analysis for RNA-seq data can be rather complicated. This is due to the extremely

large dynamic range of digital gene expression and the fact that the more highly expressed genes tend to exhibit much higher variability than weakly expressed genes. The most appropriate manner in which the results of a co-expression analysis for RNA-seq data should be displayed is still an open research question. In this manuscript, we follow the visualization approach that is conceptually similar to what has been suggested first in [1]. In this approach, bar widths correspond to the estimated proportions for the corresponding cluster $\hat{\pi}_k$. The proportion of reads that is attributed to each cell line in each cluster is represented by the corresponding colored segment within each bar. More specifically, let $y_{ijq}$ be the read of the $i$th gene for the $q$th replicate of the $j$th condition where $j = 1, \ldots, J$, $i = 1, \ldots, I_k$, $q = 1, \ldots, 3$ and $I_k$ is the number of genes in the $k$th cluster. Then, the height of the vertical bin that corresponds to the $j$th condition in $k$th cluster is

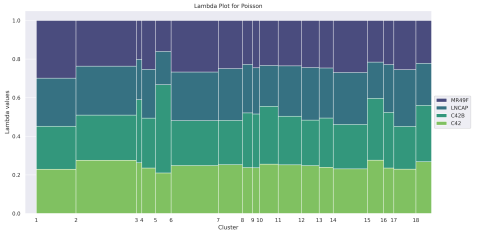
$$\lambda_{jk} = \frac{\sum_{i=1}^{i_k} \sum_{q=1}^{3} y_{ijq}}{\sum_{k=1}^{K} \sum_{i=1}^{i_k} \sum_{q=1}^{3} y_{ijq}}.$$

The results are given in the Figure (2).

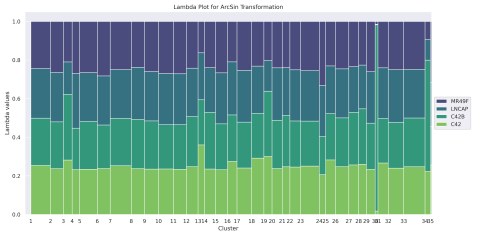
In order to determine the biological relevance of our clustering results, we performed gene ontology (GO) enrichment analyses on genes in clusters. This analysis identifies enrichment of biological processes amongst genes in clusters, thus investigating whether genes that our methods identify as co-expressed encode proteins that perform similar biological functions. In all enrichment analyses, all genes used in the differential expression analysis serve as the background gene universe, i.e. all genes remaining after filtering those with lower than 1.5 CPM expression observed. Overall, the npMSL method identifies more biologically relevant clusters than any of the other three methods. In the case of a 17 cluster npMSL solution, 14 (82.4% of clusters) show enrichment of biological process GO terms. For example, cluster 1 is associated with terms related to translation initiation (GO:0006413) such as cotranslational protein targeting to membrane (GO:0006613) and SRP-dependent cotranslational protein targeting to membrane (GO:0006614). Cluster 6 is associated with non-coding RNA processing (GO:0034470). Cluster 10 is heavily enriched for mRNA catabolic processes(GO:0006402) and cluster 14 is involved in tRNA processing (GO:0008033) including tRNA modifications (GO:0006400) and tRNA metabolic processes (GO:0006399). In comparison, the arcSin transformation method resulted in 12 out of 35 clusters (34.2% of the clusters) showing a significant enrichment of GO terms and the clusters derived from the Poisson method results in only 5 out of 18 clusters (27.8% of clusters) with enrichment of GO terms. The performance of mclust on the prostate cancer dataset is also quite poor. Only 9 out of 17 clusters (52.9%) that mclust identifies in the prostate cancer dataset



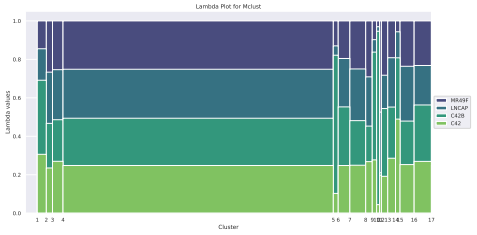

(a) Cluster behavior for the prostate cancer cell line dataset:npMSL




(b) Cluster behavior for the prostate cancer cell line dataset:Poisson method




(c) Cluster behavior for the prostate cancer cell line dataset:Transformation Method




(d) Cluster behavior for the prostate cancer cell line dataset:Mclust Method

Figure 2: Four cluster visualization plots for the prostate cancer cell line dataset

show significant enrichment of biological processes and those that do show an enrichment of much more general biological process terms.

Finally, we note that the analysis was run on a computer with 500 GB RAM with the AMD EPYC 7502 32-Core Processor CPU. The first (initialization) stage was based on the small EM algorithm of [26] and took 10 minutes with 1000 starting points. Next, we found out that it takes around 70 seconds per iteration of the npMSL method to obtain a clustering solution, with the total number of iterations being 500.

# 5 Discussion

In this manuscript, we proposed a possible method of discovering gene co-expression networks in RNA-seq data. The suggested method entails the use of a rigorous framework for parameter estimation, based on MM (Maximization-Minorization) algorithms. The model we use is distinct from most models routinely used in the clustering of digital gene expression profiles since it is a nonparametric one. More specifically, we assume that, conditional on knowing the cluster an observations has been generated from, biological samples (corresponding to marginal distributions) are independent. Moreover, each marginal distribution is not assumed to belong to any predetermined distributional family.

Conceptually, our proposal is closest to that of [1] that also uses model with conditionally independent biological samples but imposes a Poisson restriction on marginal distributions. The Poisson assumption may not be very realistic in practice since it does not account for the overdispersion routinely observed in the RNA-seq data. The attempts to handle the overdispersion problem from the parametric viewpoint typically concentrate on proposing specific distribution to handle it such as e.g. negative binomial in [2] or the multivariate Poisson-lognormal in [4]. By comparison, our method is a more general one since it imposes no specific assumptions on the marginal distributions of individual biological samples. Moreover, fitting per-gene dispersion parameters using the experimental data with multiple conditions is typically rather difficult due to a small number of replicates available in such datasets. Our proposed method avoids this problem altogether by avoiding the parametric framework for marginal distributions. The use of nonparametric multivariate clustering in bioinformatics has so far been extremely limited [8]; to the best of our knowledge, these models have not been used for the discovery of gene co-expression networks before.

We also demonstrate the behavior of our method on two real datasets: a mouse tissue dataset and a human prostate cancer cell line dataset.

| | mclust | npMSL | Poisson | Transformation |
|---|---|---|---|---|
| mclust | 1 | 0.044007 | 0.023163 | 0.063926 |
| npMSL | 0.044007 | 1 | 0.089651 | 0.123015 |
| Poisson | 0.023163 | 0.089651 | 1 | 0.224809 |
| Transformation | 0.063926 | 0.123015 | 0.224809 | 1 |

Table 1: ARI values for the mouse tissue dataset

| | mclust | npMSL | Poisson | Transformation |
|---|---|---|---|---|
| mclust | 1 | 0.084860 | 0.021193 | 0.026292 |
| npMSL | 0.084860 | 1 | 0.013246 | 0.026827 |
| Poisson | 0.021193 | 0.013246 | 1 | 0.110776 |
| Transformation | 0.026292 | 0.026827 | 0.110776 | 1 |

Table 2: ARI values for the human prostate cancer cell line dataset

The results are compared to those obtained by applying the Poisson method of [1], the transformation method of [9] and the combination method of [17] to those same datasets. In both cases, the npMSL method seems to identify a number of biologically meaningful clusters that is at least comparable to the number produced by other methods; in the human prostate cancer cell-line dataset case, it outperforms both alternative methods. For both datasets, the performance of the methods has also been compared using the adjusted Rand index (ARI) [27]. The results are summarized in Tables (1) and (2). The most salient feature of this summaries is that the clustering produced by the nonparametric npMSL method is very different from those produced by all three alternative methods. The difference is particularly striking in the case of the human prostate cancer cell line dataset where the largest ARI coefficient does not exceed 0.08. At the same time, however, the npMSL method performs much better than the other three methods in terms of GO enrichment analysis when it comes to this dataset. This suggests that this method uncovered practically important clusters that couldn't have been uncovered by other methods. In other words, these seem to represent different solutions of a problem. We would like also to note here that in the only case we are aware of where a researcher tried to applied a nonparametric method to clustering of the RNA-seq data the conclusion was a rather similar one - the clustering produced has been quite different from those produced by parametric methods [28].

The suggested model does not take into account possible dependence between biological samples because it enforces the conditional

independence between them. The natural next step in this line of research is to try to lift this assumption while preserving the general nonparametric nature of marginal distributions in our proposed model. One way this can be done is by considering explicit dependence structures between biological samples modeled using the so-called copula functions [28]. A useful direction of future research will be the introduction of specific multivariate copulas such as e.g. Gaussian copula to model the dependence between biological samples in RNA-seq data. Note that this approach allows a researcher to model any type of dependence and not just correlation which is often claimed to be the benefit of using certain so-called hidden layer approaches [3]. Our research in this direction is ongoing.

# 6 Funding

This work was supported in part by funds from the Purdue University Center for Cancer Research[P30CA082709]; the IU Comprehensive Cancer Center [P30ca082709]; and the Walther Cancer Foundation.

# References

[1] Andrea Rau, Cathy Maugis-Rabusseau, Marie-Laure Martin-Magniette, and Gilles Celeux. Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. *Bioinformatics*, 31(9):1420–1427, 2015.

[2] Yaqing Si, Peng Liu, Pinghua Li, and Thomas P Brutnell. Model-based clustering for RNA-seq data. *Bioinformatics*, 30(2):197–205, 2014.

[3] Anjali Silva, Steven J Rothstein, Paul D McNicholas, and Sanjeena Subedi. A multivariate Poisson-log normal mixture model for clustering transcriptome sequencing data. *BMC Bioinformatics*, 20(1):1–11, 2019.

[4] Sanjeena Subedi and Ryan P Browne. A family of parsimonious mixtures of multivariate Poisson-lognormal distributions for clustering multivariate count data. *Stat*, 9(1):e310, 2020.

[5] Tatiana Benaglia, Didier Chauveau, and David R Hunter. An EM-like algorithm for semi-and nonparametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics*, 18(2):505–526, 2009.

[6] Michael Levine, David R Hunter, and Didier Chauveau. Maximum smoothed likelihood for multivariate mixtures. *Biometrika*, 98(2):403–416, 2011.

[7] Didier Chauveau, David R Hunter, and Michael Levine. Semi-parametric estimation for conditional independence multivariate finite mixture models. *Statistics Surveys*, 9:1–31, 2015.

[8] Benedict Anchang, Mary T Do, Xi Zhao, and Sylvia K Plevritis. Ccast: a model-based gating strategy to isolate homogeneous subpopulations in a heterogeneous population of single cells. *PLoS computational biology*, 10(7):e1003664, 2014.

[9] Andrea Rau and Cathy Maugis-Rabusseau. Transformation and model choice for RNA-seq co-expression analysis. *Briefings in Bioinformatics*, 19(3):425–436, 2018.

[10] Zhou Shen, Michael Levine, and Zuofeng Shang. An MM algorithm for estimation of a two component semiparametric density mixture with a known component. *Electronic Journal of Statistics*, 12(1):1181–1209, 2018.

[11] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[12] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[13] Tatiana Benaglia, Didier Chauveau, and David R Hunter. Bandwidth selection in an em-like algorithm for nonparametric multivariate mixtures. In *Nonparametric Statistics And Mixture Models: A Festschrift in Honor of Thomas P Hettmansperger*, pages 15–27. World Scientific, 2011.

[14] Hiroyuki Kasahara and Katsumi Shimotsu. Non-parametric identification and estimation of the number of components in multivariate mixtures. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 97–111, 2014.

[15] Caleb Kwon and Eric Mbakop. Estimation of the number of components of nonparametric multivariate finite mixture models. *The Annals of Statistics*, 49(4):2178–2205, 2021.

[16] Jonathan Li and Andrew Barron. Mixture density estimation. *Advances in neural information processing systems*, 12, 1999.

[17] Jean-Patrick Baudry, Adrian E Raftery, Gilles Celeux, Kenneth Lo, and Raphael Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical statistics*, 19(2):332–353, 2010.

[18] Luca Scrucca, Michael Fop, T Brendan Murphy, and Adrian E Raftery. mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R journal*, 8(1):289, 2016.

[19] Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

[20] James G Booth, George Casella, and James P Hobert. Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):119–139, 2008.

[21] Dawn B Woodard and Moises Goldszmidt. Online model-based clustering for crisis identification in distributed computing. *Journal of the American Statistical Association*, 106(493):49–60, 2011.

[22] Daphne Tsoucas, Rui Dong, Haide Chen, Qian Zhu, Guoji Guo, and Guo-Cheng Yuan. Accurate estimation of cell-type composition from gene expression data. *Nat Commun*, 10(1):2975, Jul 2019.

[23] Elia Farah, Chaohao Li, Lijun Cheng, Yifan Kong, Nadia Atallah Lanman, Pete Pascuzzi, Gabrielle Renee Lorenz, Yanquan Zhang, Nihal Ahmad, Lang Li, Timothy Ratliff, and Liu Xiaoqi. Notch signaling is activated in and contributes to resistance in enzalutamide-resistant prostate cancer cells. *Journal of Biological Chemistry*, 294(21):8543–8554, 2019.

[24] Wenpin Hou, Zhicheng Ji, Hongkai Ji, and Stephanie C Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome biology*, 21(1):1–30, 2020.

[25] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725, 2000.

[26] Jean-Patrick Baudry and Gilles Celeux. Em for mixtures: Initialization requires special care. *Statistics and computing*, 25:713–726, 2015.

[27] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[28] Gildas Mazo and Yaroslav Averyanov. Constraining kernel estimators in semiparametric copula mixture models. *Computational Statistics & Data Analysis*, 138:170–189, 2019.

# 7 Appendix

## 7.1 Mouse tissue dataset

The other dataset that we analyze is the one where the bulk RNA-seq was performed to profile the gene expression profiles in lung, kidney, liver, and small intestine tissues from six- to ten-week-old male C57BL/6J mice. As gene expression data is highly tissue-specific, data such as these can allow for the identification of tissue-specific expression modules. Two biological replicates are present for each tissue type

and separate mice were used for each replicate. Observations in these datasets are commonly referred to as "counts" where a count is the number of reads that align to a particular feature(gene). Filtering at the level of 1.5 CPM as described above resulted in a file containing $16,512$ genes. These data are available through the Gene Expression Omnibus (GEO) repository through accession number GSE124419. As before, we applied a logarithmic transform to the data as a first step.

Using again the method of selecting cluster numbers that is suggested in [17], we started with choosing the range from 1 to $K = 30$. As is the case with the human prostate cancer dataset, in a typical run, one needs about $29 - 30$ clusters with VVI or VVE covariance matrix structure. By deploying the automated strategy of the cluster number selection we found out that, in a typical run of the procedure, one ends up with 16-18 clusters as a possible suggestion. In what follows, we use 17 as a suggested cluster number.

For comparison purposes, we again compare clustering results obtained using the npMSL method with the results obtained using the Poisson method of [1], the transformation-based method of [9] and the combination method of [17]. We used the slope heuristics method to select the optimal number of clusters for the Poisson method. This approach resulted in the choice of 19 clusters when running the slope heuristics approach over a range from 1 to 35 clusters.

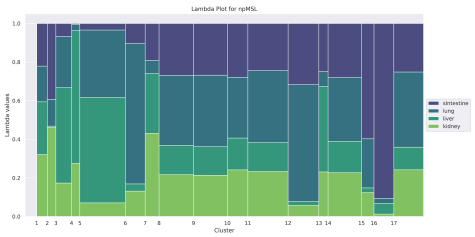
The transformation method has been used with the logit transform and the number of clusters has been selected using the ICL (Integrated Complete Likelihood) method [25]. Using this criterion over a possible range of 1 to 40 clusters, we found that 25 seemed to be the optimal number in this case.

The final comparison method that we use is a combination method of [17]. As mentioned earlier while discussing the choice of the number of clusters for the npMSL method, this approach suggests the choice of 17 clusters. Visualizations of resulting cluster solutions for all four methods are given in Figures (3a)-(3b)-(3c)-(3d).

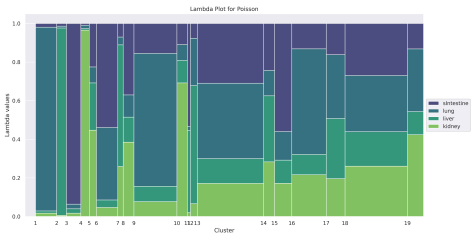
The visualization of results is done in exactly the same manner as for the human prostate cancer cell line dataset and is given in Figure (3a)-(3b)-(3c)-(3d).

Note that this approach to visualization allows us to assess the relative size of clusters with different level of expression in different tissues. For example, when one looks at the visualization produced by the npMSL method, one can see that clusters that contain mostly genes that are strongly expressed in small intestine and lungs tend to be larger than those that contain mostly genes that are expressed in liver or kidneys.

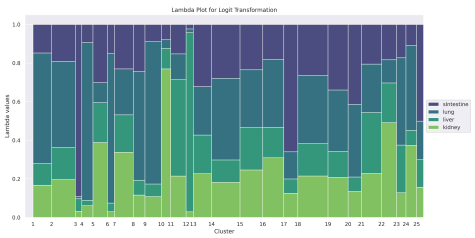
In order to determine the biological relevance of our clustering results, we performed gene ontology (GO) enrichment analyses on genes in clusters. This analysis identifies enrichment of biological processes



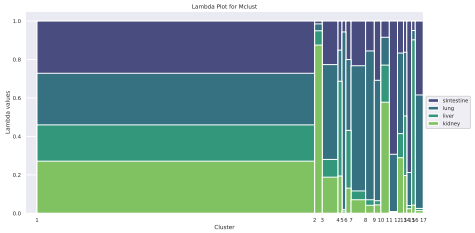

(a) Cluster behavior for the mouse tissue dataset:npMSL




(b) Cluster behavior for the mouse tissue dataset:Poisson method




(c) Cluster behavior for the mouse tissue dataset: Transformation Method




(d) Cluster behavior for the mouse tissue dataset: mclust method

Figure 3: Three cluster visualization plots for the mouse tissue dataset

amongst genes in clusters, thus investigating whether genes that our methods identify as co-expressed encode proteins that perform similar biological functions. In all enrichment analyses, all genes used in the differential expression analysis serve as the background gene universe, i.e. all genes remaining after filtering those with lower than 1.5 CPM expression observed. All the clustering methods used identify a somewhat larger number of biologically relevant clusters when applied to this cell line dataset compared to the human prostate cancer dataset. Out of the 17 clusters identified by the npMSL method, 16 of them (94.1%) show significant enrichment of biological processes. The clusters identified are associated with highly specific biological processes. For example, cluster 2 involves genes that function in water homeostasis (GO:0030104). Cluster 12 is clearly associated with processes involved in the adaptive immune response (GO:0002250), and also includes specific terms such as regulation of T cell activation (GO:0042110) and B cell activation (GO:0042113). Cluster 16 on the other hand involves GO terms related to the innate immune response, such as the innate immune response in mucosa (GO:0002227) and the defense response to Gram-positive bacterium (GO:0050830). For this dataset, the Poisson method produces 19 clusters that are all associated with statistically enriched GO terms while the logit transformation method results in the lowest performance, with 22 out of 25 clusters (88%) enriched for biological processes. Finally, we also look at the GO enrichment analysis of the 17 cluster solution obtained using the combination method of [17]. This method with 17 clusters chosen produces a clustering solution that is quite different from other clusters: it suggests one very large clusters with the rest of clusters being quite small. Nevertheless, all 17 clusters thus produced are enriched for biological processes, most of which are quite specific, suggesting that clusters indeed contain genes that perform distinct biological functions.

For this dataset, the first (initialization) stage was again based on the small EM algorithm and took 15 minutes with 1000 starting points. Next, we found out that it takes around 330 seconds per iteration of the npMSL method to obtain a clustering solution, with the total number of iterations being 500.