Co-expression analysis of RNA-seq data with the coseq package

Andrea Rau[1] and Cathy Maugis-Rabusseau

[1]andrea.rau@jouy.inra.fr

coseq version 0.1.10

Abstract

This vignette illustrates the use of the *coseq* package through an example of a co-expression analysis using the RNA-seq data of mouse embryonic neocortex from [1]. The *coseq* package is devoted to the co-expression analysis of sequencing data. It contains the new strategy based on Gaussian mixture models on transformed profiles [see 2, for more details] and the Poisson mixture models developed in *HTSCluster* [3]. For a full presentation of the statistical methods, please see our papers [3, 2].

Contents

1 Input data

In this vignette, we will work with the gene-level read counts data of mouse embryonic neocortex from [1]. They studied the expansion of the neocortex in five embryonic (day 14.5) mice by analyzing the transcriptome of the ventricular zone (VZ), subventricular zone (SVZ), and cortical plate (CP) using RNA-seq. Laser-capture microdissection, RNA isolation and cDNA library preparation, and RNA sequencing and quantification are described in the Supplementary Materials of [1]. In our work, raw read counts for this study were downloaded on December 23, 2015 from the Digital Expression Explorer (DEE) [4] using associated SRA accession number SRP013825, and run information was downloaded using the SRA Run Selector. The raw read counts for 8962 genes (after filtering those with mean < 50) are here considered. We begin by loading the necessary packages, data, and phenotypic information for the analysis.

```
> library(coseq)
> counts<-read.table("http://www.math.univ-toulouse.fr/~maugis/coseq/Fietz_mouse_counts.txt",header=T)
> conds<-c(rep(1,5),rep(2,5),rep(3,5))
> head(counts)
```

	CP	CP.1	CP.2	CP.3	CP.4	SVZ	SVZ.1	SVZ.2	SVZ.3	SVZ.4	VZ	VZ.1	VZ.2	VZ.3	VZ.4
1	876	965	575	491	389	892	904	563	391	368	1125	883	871	472	696
2	23	25	13	4	17	52	30	37	37	14	204	138	165	69	137
3	131	67	44	38	99	99	92	131	70	37	109	148	107	73	192
4	884	1107	666	465	359	675	710	396	326	237	505	352	364	210	261
5	240	284	158	137	97	78	110	46	34	23	110	96	93	47	71
6	337	431	211	159	131	176	259	137	102	74	297	216	199	92	155

2 Identifying co-expressed genes with Gaussian mixtures on transformed data

2.1 Model description

The following description closely follows that provided in our main paper [2].

Let $\mathbf{y}$ be the $n \times q$ matrix of read counts where y_{ij}, represents to the raw read count for biological entity i ($i = 1, \ldots, n$) of biological sample j ($j = 1, \ldots, q$). For simplicity, in this work we typically refer to the entities i as genes, although the generality of the following discussion holds for other entities of interest (proteins, exons, etc). Each biological sample is typically associated with one or more experimental conditions (e.g., tissue, treatment,). Finally, let $\mathbf{y}_i$ be the q-dimensional vector of raw count values across all biological samples for gene i and $y_{i\cdot} = \sum_j y_{ij}$.

We propose to use *normalized expression profiles*, that is, the proportion of normalized reads observed for gene i with respect to the total observed for gene i across all samples: $\mathbf{p}_i = (p_{i1}, \ldots, p_{iq})$ where

$$p_{ij} = \frac{y_{ij}/s_j}{\sum_j y_{ij}/s_j},$$

$(t_j)_j$ are the scaling normalization factors for raw library sizes (the TMM normalization method by default), $\ell_j = y_{\cdot j} t_j$ and $s_j = q\, \ell_j / \sum_j \ell_j$. Since the coordinates of $\mathbf{p}_i$ are linearly dependent, the direct adjustement of a Gaussian mixture distribution is problematic. For this reason, we first consider two separate transformations of the profiles p_{ij}:

- arcsin transformation: $\arcsin\left(\sqrt{p_{ij}}\right)$
- logit transformation: $\log_2\left(\frac{p_{ij}}{1-p_{ij}}\right)$

Second, the distribution of the transformed normalized expression profiles is modelled by a general multidimensional Gaussian mixture

$$f(.|\theta_K) = \sum_{k=1}^{K} \pi_k \phi(.|\mu_k, \Sigma_k) \tag{1}$$

where $\theta_K = (\pi_1, \ldots, \pi_{K-1}, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$, $\pi = (\pi_1, \ldots, \pi_K)$ are the mixing proportions and $\phi(.|\mu_k, \Sigma_k)$ is the q-dimensional Gaussian density function with mean μ_k and covariance matrix Σ_k.

To estimate mixture parameters θ_K by computing the maximum likelihood estimate (MLE), an Expectation-Maximization (EM) algorithm is performed. It is implemented in the *Rmixmod* package. For model selection (i.e. the choice of the number of clusters K) we make use of the Integrated Completed Likelihoood (ICL) criterion.

Let $\hat{t}_{ik}$ be the conditional probability that observation i arises from the kth component of the mixture $f(.|\hat{\theta}_{\hat{K}})$. Finally, each observation i is assigned to the component maximizing the conditional probability $\hat{t}_{ik}$ i.e., using the so-called maximum a posteriori (MAP) rule.

2.2 Co-expression analysis of mouse embryonic neocortex data

In this example, we perform a single run of coseq for the arcsin transformation and Gaussian mixture model for $K = 2, \ldots, 40$ clusters. In order to reduce computation time, we use a parallel execution using *BiocParallel*.

```
> ## ATTENTION: this code is somewhat long to run
> set.seed(12345)
> runArcsin <- coseq(counts, K=2:40, norm="TMM", model="Normal",
+                    transformation="arcsin", parallel=T)
```

The results are available as follows:

```
> load(url("http://www.math.univ-toulouse.fr/~maugis/coseq/ArcsinGmouse.RData"))
```

A built-in summary command allows a text-based overview of the selected number of clusters, the number of genes in each cluster, the number of genes with maximum conditional probabilities greater than 90%, the number of genes in each cluster with maximum conditional probabilities greater than 90%, and the estimated values of the means $\hat{\mu}_k$ and $\hat{\pi}$. This is available as follows:

```
> summary(runmouseArcsin)

*********************************************
Model: Normal
Transformation: arcsin
*********************************************
Clusters fit: 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,34,36,37
Clusters with errors: 31,33,35,39
Selected number of clusters via ICL: 12
ICL of selected model: -707646
*********************************************
Cluster sizes:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
      1020        387        306       1389        448       1176        323
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12
       484        695        858        869       1001

Number of observations with MAP > 0.90 (% of total):
6452 (72%)

Number of observations with MAP > 0.90 per cluster (% of total per cluster):
 Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Cluster 7
 734       340       259       861       425       798       266
 (71.96%)  (87.86%)  (84.64%)  (61.99%)  (94.87%)  (67.86%)  (82.35%)
 Cluster 8 Cluster 9 Cluster 10 Cluster 11 Cluster 12
 394       589       588        577        621
 (81.4%)   (84.75%)  (68.53%)   (66.4%)    (62.04%)

Mu:
              CP  CP.1  CP.2  CP.3  CP.4   SVZ SVZ.1 SVZ.2 SVZ.3 SVZ.4    VZ
Cluster 1  0.233 0.231 0.229 0.231 0.236 0.253 0.249 0.253 0.254 0.253 0.294
Cluster 2  0.395 0.412 0.410 0.413 0.376 0.166 0.197 0.161 0.150 0.163 0.108
Cluster 3  0.269 0.269 0.255 0.256 0.267 0.237 0.244 0.233 0.224 0.221 0.247
Cluster 4  0.264 0.267 0.265 0.263 0.259 0.258 0.260 0.256 0.256 0.254 0.264
Cluster 5  0.136 0.094 0.097 0.092 0.172 0.167 0.143 0.178 0.168 0.164 0.402
Cluster 6  0.258 0.258 0.255 0.249 0.251 0.259 0.259 0.253 0.253 0.250 0.273
Cluster 7  0.209 0.208 0.207 0.209 0.212 0.326 0.315 0.333 0.340 0.333 0.209
Cluster 8  0.333 0.342 0.342 0.347 0.320 0.267 0.284 0.264 0.263 0.268 0.130
Cluster 9  0.206 0.198 0.193 0.193 0.217 0.230 0.223 0.229 0.227 0.225 0.334
Cluster 10 0.278 0.281 0.285 0.289 0.276 0.270 0.271 0.272 0.274 0.274 0.223
Cluster 11 0.305 0.313 0.310 0.307 0.292 0.255 0.266 0.250 0.248 0.249 0.215
Cluster 12 0.247 0.247 0.254 0.256 0.252 0.266 0.258 0.268 0.272 0.273 0.261
            VZ.1  VZ.2  VZ.3  VZ.4
```

```
Cluster 1  0.294 0.294 0.298 0.292
Cluster 2  0.109 0.112 0.106 0.126
Cluster 3  0.252 0.250 0.235 0.256
Cluster 4  0.262 0.262 0.258 0.261
Cluster 5  0.408 0.409 0.402 0.400
Cluster 6  0.273 0.269 0.265 0.269
Cluster 7  0.208 0.214 0.210 0.212
Cluster 8  0.129 0.133 0.125 0.139
Cluster 9  0.335 0.335 0.335 0.332
Cluster 10 0.223 0.226 0.226 0.225
Cluster 11 0.216 0.215 0.208 0.215
Cluster 12 0.262 0.263 0.270 0.262

Pi:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
     0.113      0.043      0.036      0.154      0.050      0.136      0.037
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12
     0.054      0.078      0.094      0.096      0.108
```

A built-in plot command allows a graphical overview of the results:

```
> plot(runmouseArcsin,order=T,conds=conds,average_over_conds=T)
```

Respectively, the different plots correspond to:

- the plot of the log-likelihood versus the number of clusters (see Figure 1, left)
- the plot of ICL versus the number of clusters (see Figure 1, right)
- line plots of profiles in each cluster (average values within each condition identified by conds, see Figure 2)
- boxplots of profiles in each cluster (average values within each condition identified by conds, see Figure 3)
- boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster (see Figure 4)
- number of observations with a maximum conditional probability greater than threshold per cluster (see Figure 5)
- a histogram of maximum conditional probabilities of cluster membership for all genes (see Figure 5)

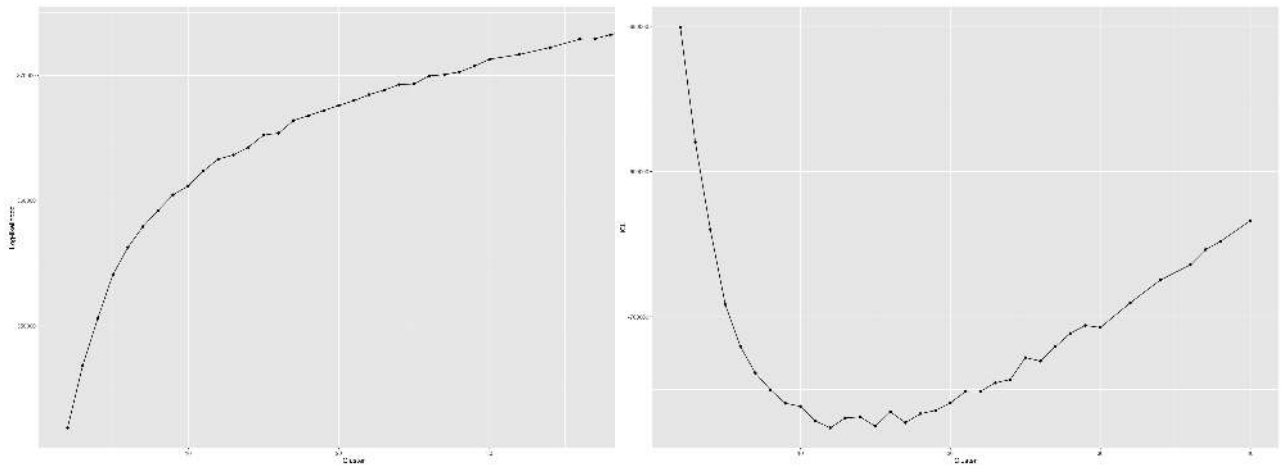

Figure 1: Visualization of the loglikelihood (left) and the ICL criterion (right) versus the number of clusters.

Each of these graphs are also available by individual command as follows:

```
> plot(runmouseArcsin,graphs="logLike")
> plot(runmouseArcsin,graphs="ICL")
> plot(runmouseArcsin,graphs="profiles",conds=conds,average_over_conds=T)
```

```
> plot(runmouseArcsin,graphs="boxplots",conds=conds,average_over_conds=T)
> plot(runmouseArcsin,graphs="probapost_boxplots")
> plot(runmouseArcsin,graphs="probapost_barplots",order=T)
> plot(runmouseArcsin,graphs="probapost_histogram")
```

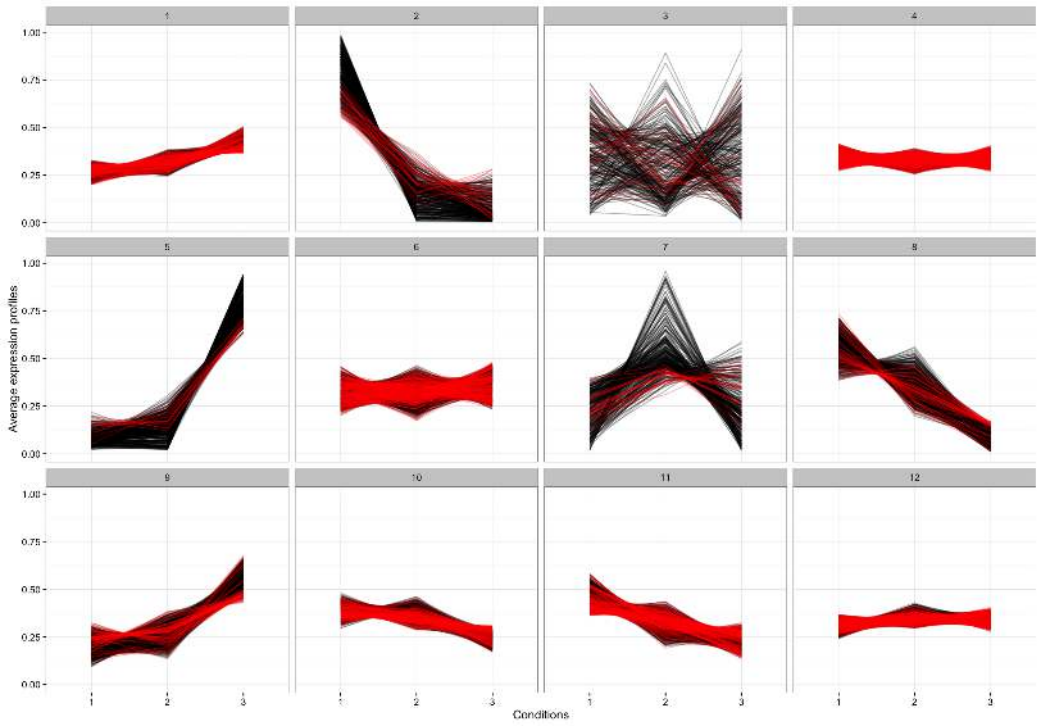

Figure 2: Visualization of the gene profiles in each cluster (average values within each condition identified by conds).

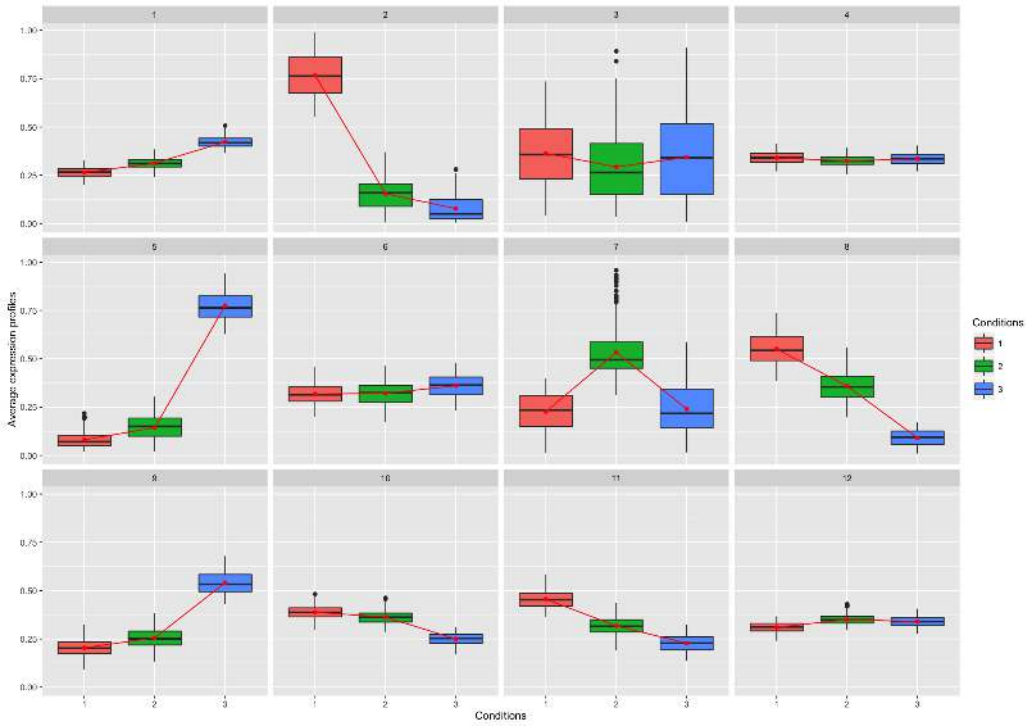

Figure 3: Visualization of boxplots of gene profiles in each cluster (average values within each condition identified by conds).

We may also examine a histogram of maximum conditional probabilities of cluster membership for all genes (Figure 5), as well as boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster (Figure 4). These plots help to evaluate the degree of certitude accorded by the model in assigning genes to clusters, as well as whether some clusters are attribued a greater degree of uncertainty than others.

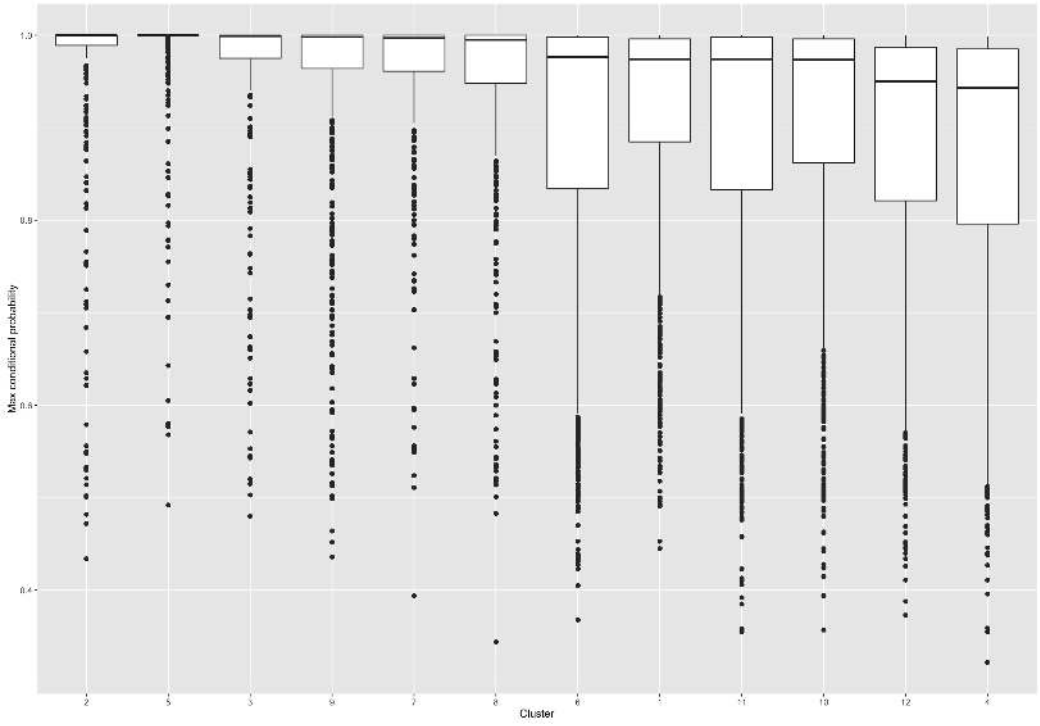

Figure 4: Boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster.

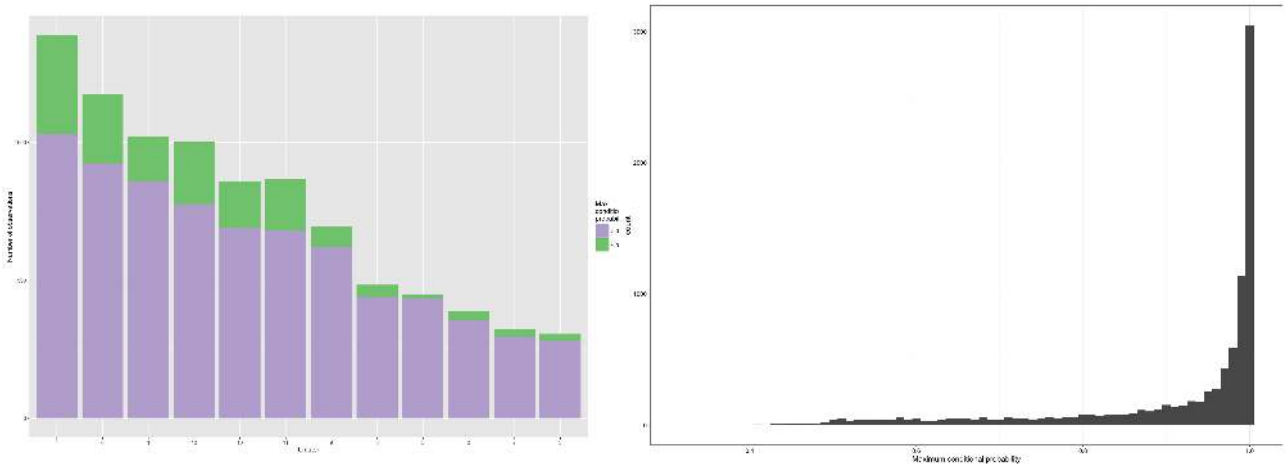

Figure 5: Left: Number of observations with a maximum conditional probability greater than threshold per cluster. Right:Histogram of maximum conditional probabilities of cluster membership.

The cluster labels and conditional probabilities of cluster membership assigned to each gene may be accessed using the following code:

```
> probaPost <- runmouseArcsin$results$ICL.results$probaPost
> labelG<-apply(probaPost,1,which.max)
```

To compare the clusterings obtained for some values of K, the function compareARI can be used:

```
> compareARI(runmouseArcsin,K=11:14)
```

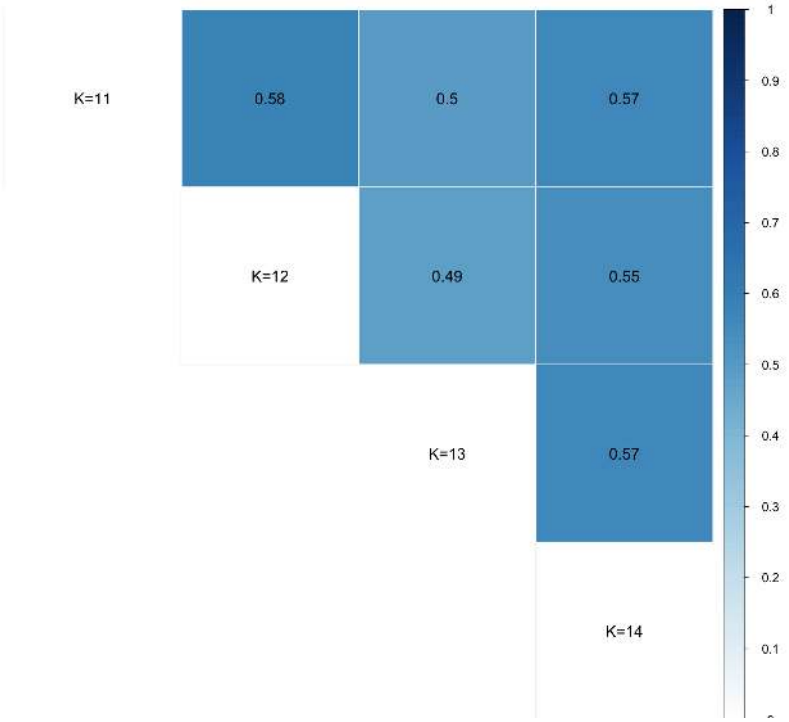

Figure 6: Visualization of the calculated pairwise ARI values for $K = 11$ to 14.

In the same way, the results for the logit transformation on the RNA-seq data of mouse embryonic neocortex are available as follows:

```
> load(url("http://www.math.univ-toulouse.fr/~maugis/coseq/LogitGmouse.RData"))
```

3 Identifying co-expressed genes with Poisson mixtures

3.1 Model description

The following description closely follows that provided in our main paper [3].

Let Y_{ijl} be the random variable corresponding to the digital gene expression measure (DGE) for biological entity i ($i = 1, \ldots, n$) of condition j ($j = 1, \ldots, d$) in biological replicate l ($l = 1, \ldots, r_j$), with y_{ijl} being the corresponding observed value of Y_{ijl}. Let $q = \sum_{j=1}^{d} r_j$ be the total number of variables (all replicates in all conditions) in the data, such that $\mathbf{y} = (y_{ijl})$ is the $n \times q$ matrix of the DGE for all observations and variables, and $\mathbf{y}_i$ is the q-dimensional vector of DGE for all variables of observation i. We use dot notation to indicate summations in various directions, e.g., $y_{\cdot jl} = \sum_i y_{ijl}$, $y_{i\cdot\cdot} = \sum_j \sum_l y_{ijl}$, and so on.

3.1.1 Poisson mixture model

To cluster RNA-seq data, we consider a model-based clustering procedure based on mixture of Poisson distributions. The data $\mathbf{y}$ are assumed to come from K distinct subpopulations (clusters), each of which is modeled separately:

$$f(\mathbf{y}; K, \boldsymbol{\Psi}_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_i; \boldsymbol{\theta}_{ik})$$

where $\boldsymbol{\Psi}_K = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\theta}')'$, $\boldsymbol{\theta}'$ contains all of the parameters in $\{\boldsymbol{\theta}_{ik}\}_{i,k}$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ are the mixing proportions, with $\pi_k \in (0, 1)$ for all k and $\sum_{k=1}^{K} \pi_k = 1$. Samples are assumed to be independent conditionally on the components:

$$f_k(\mathbf{y}_i; \boldsymbol{\theta}_{ik}) = \prod_{j=1}^{d} \prod_{l=1}^{r_j} \mathcal{P}(y_{ijl}; \mu_{ijlk}),$$

where $\mathcal{P}(\cdot;\mu_{ijlk})$ denotes the standard Poisson probability mass function with mean μ_{ijlk}.

Each mean μ_{ijlk} is parameterized by

$$\mu_{ijlk} = w_i s_{jl} \lambda_{jk}$$

where $w_i = y_{i\cdot\cdot}$ corresponds to the overall expression level of observation i (e.g., weakly to strongly expressed) and s_{jl} represents the normalized library size for replicate l of condition j, such that $\sum_{j,l} s_{jl} = 1$. These normalization factors take into account the fact that the number of reads expected to map to a particular gene depends not only on its expression level, but also on the library size (overall number of mapped reads) and the overall composition of the RNA population being sampled. We note that $\{s_{jl}\}_{j,l}$ are estimated from the data prior to fitting the model, and like the overall expression levels w_i, they are subsequently considered to be fixed in the Poisson mixture model. Finally, the unknown parameter vector $\boldsymbol{\lambda}_k = (\lambda_{1k}, \ldots, \lambda_{dk})$ corresponds to the clustering parameters that define the profiles of the genes in cluster k across all biological conditions.

3.1.2 Inference

To estimate mixture parameters $\boldsymbol{\Psi}_K = (\boldsymbol{\pi}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K)$ by computing the maximum likelihood estimate (MLE), an Expectation-Maximization (EM) algorithm is considered. After initializing the parameters $\boldsymbol{\Psi}_K^{(0)}$ and $\mathbf{z}^{(0)}$ by a so-called Small-EM strategy, the E-step at iteration b corresponds to computing the conditional probability that an observation i arises from the kth component for the current value of the mixture parameters:

$$t_{ik}^{(b)} = \frac{\pi_k^{(b)} f_k(\mathbf{y}_i; \boldsymbol{\theta}_{ik}^{(b)})}{\sum_{m=1}^{K} \pi_m^{(b)} f_m(\mathbf{y}_i; \boldsymbol{\theta}_{im}^{(b)})}$$

where $\boldsymbol{\theta}_{ik}^{(b)} = \{w_i s_{jl} \lambda_{jk}^{(b)}\}_{jl}$. Then, in the M-step the mixture parameter estimates are updated to maximize the expected value of the completed likelihood, which leads to weighting the observation i for group k with the conditional probability $t_{ik}^{(b)}$. Thus,

$$\pi_k^{(b+1)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}^{(b)} \quad \text{and} \quad \lambda_{jk}^{(b+1)} = \frac{\sum_{i=1}^{n} t_{ik}^{(b)} y_{ij\cdot}}{s_{j\cdot} \sum_{i=1}^{n} t_{ik}^{(b)} y_{i\cdot\cdot}},$$

since $w_i = y_{i\cdot\cdot}$. Note that at each iteration of the EM algorithm, we obtain that $\sum_{j=1}^{d} \lambda_{jk}^{(b)} s_{j\cdot} = 1$. Thus $\lambda_{jk}^{(b)} s_{j\cdot}$ can be interpreted as the proportion of reads that are attributed to condition j in cluster k, after accounting for differences due to library size; this proportion is shared among the replicates of condition j according to their respective library sizes s_{jl}.

3.1.3 Model selection

For model selection (i.e., the choice of the number of clusters K), we make use of the so-called *slope heuristics*, which is a data-driven method to calibrate a penalized criterion that is known up to a multiplicative constant. Briefly, in our context the penalty is assumed to be proportional to the number of free parameters ν_K (i.e., the model dimension), such that $\text{pen}(K) \propto \kappa \nu_K$; we note that this assumption may be verified in practice. The penalty is calibrated using the *data-driven slope estimation* (DDSE) procedure available in the capushe R package [5]. This procedure directly estimates the slope of the expected linear relationship of the loglikelihood with respect to the model dimension for the most complex models (here, models with large K). Denoting the estimated slope $\hat{\kappa}$, in our context the slope heuristics consists of setting the penalty to be $2\hat{\kappa}\nu_K$. The number of selected clusters $\hat{K}$ then corresponds to the value of K minimizing the penalized criterion:

$$\text{crit}(K) = -\log f(\mathbf{y}; K, \hat{\boldsymbol{\Psi}}_K) + 2\hat{\kappa}\nu_K.$$

Finally, we note that capushe also provides an alternative procedure for calibrating the penalty called the *dimension jump* (Djump). For more details about the DDSE and Djump approaches, see [5].

Based on $\hat{\boldsymbol{\Psi}}_{\hat{K}}$, each observation i is assigned to the component maximizing the conditional probability $\hat{t}_{ik}$ (i.e., using the so-called MAP rule).

3.2 Co-expression analysis of mouse embryonic neocortex data

We perform a single run of coseq for Poisson mixture model for $K = 2, \ldots, 60$ clusters, using the Trimmed Means of M-values (TMM) normalization [6]. In order to reduce computation time, we use a parallel execution using *BiocParallel* (thus the splitting small-EM strategy described in [3] is not used here.)

```
> ## ATTENTION: this code is somewhat long to run
> set.seed(12345)
> PMM <- coseq(y=counts, K=c(2:60), conds=conds,
+              model="Poisson",   transformation="none",
+              norm="TMM", modelChoice="DDSE",parallel=T)
```

The results of this code are available as follows:

```
> load(url("http://www.math.univ-toulouse.fr/~maugis/coseq/PMMmouse.RData"))
```

The model choice is here performed using the DDSE calibration of the slope heuristics (option *modelChoice="DDSE"* in the above code). The results associated to the model selection of DDSE may be accessed as follows:

```
> mod<-PMM$results$selected.results
```

A built-in summary command allows a text-based overview of the selected model, including the number of clusters, the model selection approach (in this case, DDSE), the number of genes in each cluster, the number of genes with maximum conditional probabilities greater than 90%, the number of genes in each cluster with maximum conditional probabilities greater than 90%, and the estimated values of $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\pi}}$. It is available as follows:

```
> summary(PMM)

*********************************************
Model: Poisson
Transformation: none
*********************************************
Clusters fit: 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34
Selected number of clusters: 36
Model selection criterion: DDSE
*********************************************
Cluster sizes:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
        62         61        363         84        546        669        189
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
       900        155        162        143        232        135        149
Cluster 15 Cluster 16 Cluster 17 Cluster 18 Cluster 19 Cluster 20 Cluster 21
       120         74        133        334        261        164        603
Cluster 22 Cluster 23 Cluster 24 Cluster 25 Cluster 26 Cluster 27 Cluster 28
       107        700        615        610         26         49        121
Cluster 29 Cluster 30 Cluster 31 Cluster 32 Cluster 33 Cluster 34 Cluster 35
        94         27        152        194        194        320         48
Cluster 36
       166

Number of observations with MAP > 0.90 (% of total):
7701 (85.9%)

Number of observations with MAP > 0.90 per cluster (% of total per cluster):
 Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Cluster 7
 61        58        315       80        481       557       174
 (98.39%)  (95.08%)  (86.78%)  (95.24%)  (88.1%)   (83.26%)  (92.06%)
 Cluster 8 Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
 673       151       153        133        190        128        141
 (74.78%)  (97.42%)  (94.44%)   (93.01%)   (81.9%)    (94.81%)   (94.63%)
 Cluster 15 Cluster 16 Cluster 17 Cluster 18 Cluster 19 Cluster 20 Cluster 21
 119        72         124        294        236        148        488
 (99.17%)   (97.3%)    (93.23%)   (88.02%)   (90.42%)   (90.24%)   (80.93%)
 Cluster 22 Cluster 23 Cluster 24 Cluster 25 Cluster 26 Cluster 27 Cluster 28
 101        520        491        511        26         49         120
 (94.39%)   (74.29%)   (79.84%)   (83.77%)   (100%)     (100%)     (99.17%)
 Cluster 29 Cluster 30 Cluster 31 Cluster 32 Cluster 33 Cluster 34 Cluster 35
 94         27         149        192        174        267        47
```

```
 (100%)     (100%)     (98.03%)   (98.97%)   (89.69%)   (83.44%)   (97.92%)
 Cluster 36
 157
 (94.58%)

lamba:
  Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6 Cluster 7
1     1.789     1.208     1.202     0.817     0.750     1.167     1.425
2     0.337     1.539     1.119     1.561     0.878     0.978     0.719
3     0.707     0.316     0.679     0.722     1.376     0.837     0.777
  Cluster 8 Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
1     0.924     1.889      0.373      0.408      0.818      1.543      1.031
2     1.026     0.919      0.764      0.401      1.226      1.143      0.606
3     1.060     0.103      1.882      2.153      1.005      0.289      1.301
  Cluster 15 Cluster 16 Cluster 17 Cluster 18 Cluster 19 Cluster 20 Cluster 21
1      0.146       1.47      0.584      1.359      0.581      0.724      0.794
2      0.184       1.42      1.170      0.940      0.839      0.550      1.029
3      2.622       0.13      1.307      0.661      1.592      1.683      1.199
  Cluster 22 Cluster 23 Cluster 24 Cluster 25 Cluster 26 Cluster 27 Cluster 28
1      1.106      1.015      1.073      0.909      0.288      0.283      2.239
2      1.369      1.059      0.893      0.884      2.868      1.365      0.446
3      0.571      0.934      1.012      1.198      0.185      1.468      0.125
  Cluster 29 Cluster 30 Cluster 31 Cluster 32 Cluster 33 Cluster 34 Cluster 35
1      2.512      0.364      1.867      0.155      1.551      1.027      0.935
2      0.141      2.112      0.687      0.535      0.927      1.187      2.003
3      0.087      0.746      0.324      2.314      0.463      0.812      0.218
  Cluster 36
1      1.364
2      1.151
3      0.476

pi:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
     0.007      0.007      0.040      0.009      0.060      0.074      0.021
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
     0.099      0.017      0.018      0.016      0.026      0.015      0.017
Cluster 15 Cluster 16 Cluster 17 Cluster 18 Cluster 19 Cluster 20 Cluster 21
     0.013      0.008      0.015      0.037      0.029      0.018      0.067
Cluster 22 Cluster 23 Cluster 24 Cluster 25 Cluster 26 Cluster 27 Cluster 28
     0.012      0.078      0.068      0.070      0.003      0.006      0.014
Cluster 29 Cluster 30 Cluster 31 Cluster 32 Cluster 33 Cluster 34 Cluster 35
     0.011      0.003      0.017      0.022      0.021      0.036      0.005
Cluster 36
     0.019
```

Here, the DDSE criterion selects a clustering with 36 clusters.

Note that the slope heuristics approach may only be applied if more than 10 models are included in the model collection (i.e., if `gmax-gmin` + 1 is greater than 10); in the case where this constraint is not met, a warning message to this effect is produced. In cases where the slope heuristics approach may be applied, it is essential to verify the diagnostic plots produced by *capushe* prior to basing inference on the selected models (see below), and a message reminding the user of this is displayed. The *capushe* package provides diagnostic plots for the slope heuristics in order to ensure that sufficiently complex models have been considered. These diagnostic plots may be accessed as follows (see Figure 7):

```
> library(capushe)
> plot(PMM$results$capushe@DDSE)
```

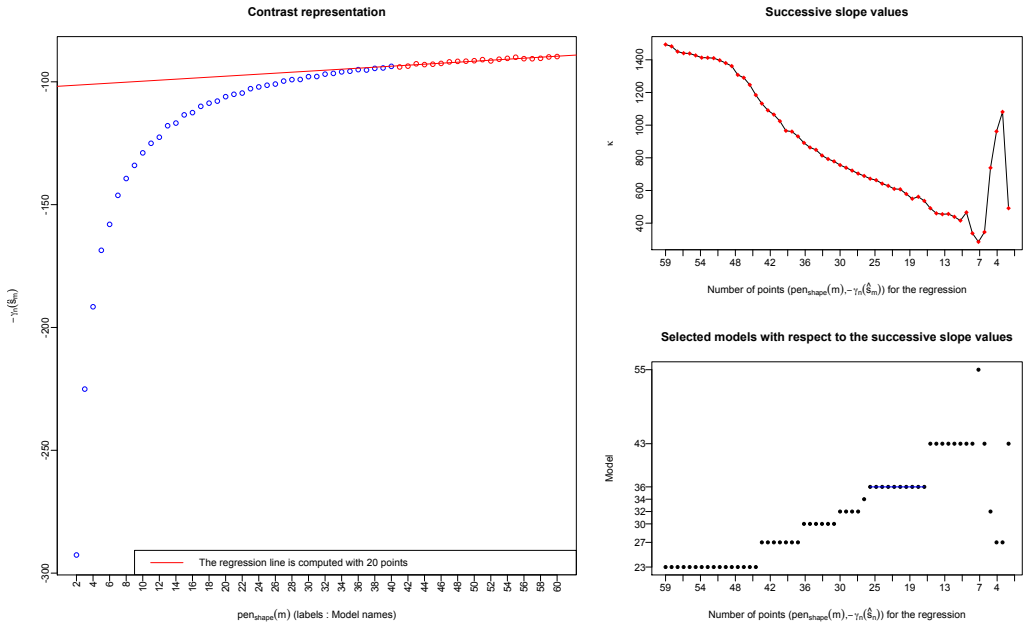

Figure 7: Diagnostic plots provided by *capushe* package for the DDSE approach; see [5] for additional information.

A built-in plot command allows a graphical overview of the results:

```
> plot(PMM,order=T,conds=conds,average_over_conds=T)
```

Respectively, the different plots correspond to:

- the plot of the log-likelihood versus the number of clusters
- the plot of ICL versus the number of clusters (this criterion is not used here)
- line plots of profiles in each cluster (average values within each condition identified by conds, see Figure 8)
- boxplots of profiles in each cluster (average values within each condition identified by conds, see Figure 9)
- boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster (see Figure 10)
- number of observations with a maximum conditional probability greater than threshold per cluster
- a histogram of maximum conditional probabilities of cluster membership for all genes (see Figure 11)
- barplots of estimated proportions of counts per condition ($\hat{\lambda}_{jk} s_j$) in each cluster for the Poisson mixture model (see Figure 12), where bar widths represent the values of $\hat{\pi}$.

Each of these graphs are also available by individual command as follows:

```
> plot(PMM,graphs="logLike")
> plot(PMM,graphs="ICL")
> plot(PMM,graphs="profiles",conds=conds,average_over_conds=T)
> plot(PMM,graphs="boxplots",conds=conds,average_over_conds=T)
> plot(PMM,graphs="probapost_boxplots")
> plot(PMM,graphs="probapost_barplots",order=T)
> plot(PMM,graphs="probapost_histogram")
> plot(PMM,graphs="lambda_barplots")
```

We may also examine a histogram of maximum conditional probabilities of cluster membership for all genes (Figure 11), as well as boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster (Figure 10). These plots help to evaluate the degree of certitude accorded by the model in assigning genes to clusters, as well as whether some clusters are attribued a greater degree of uncertainty than others.

The cluster labels and conditional probabilities of cluster membership assigned to each gene may be accessed using the following code:

```
> labelPMM <- mod$labels
> probaPost <- mod$probaPost
```

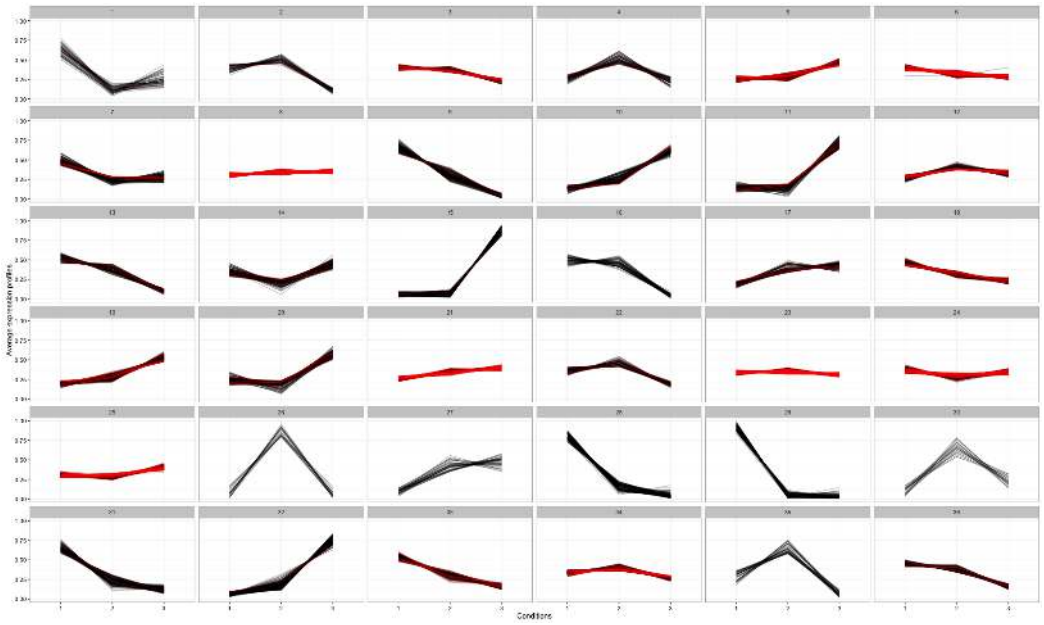

Figure 8: Visualization of the gene profiles in each cluster (average values within each condition identified by conds).

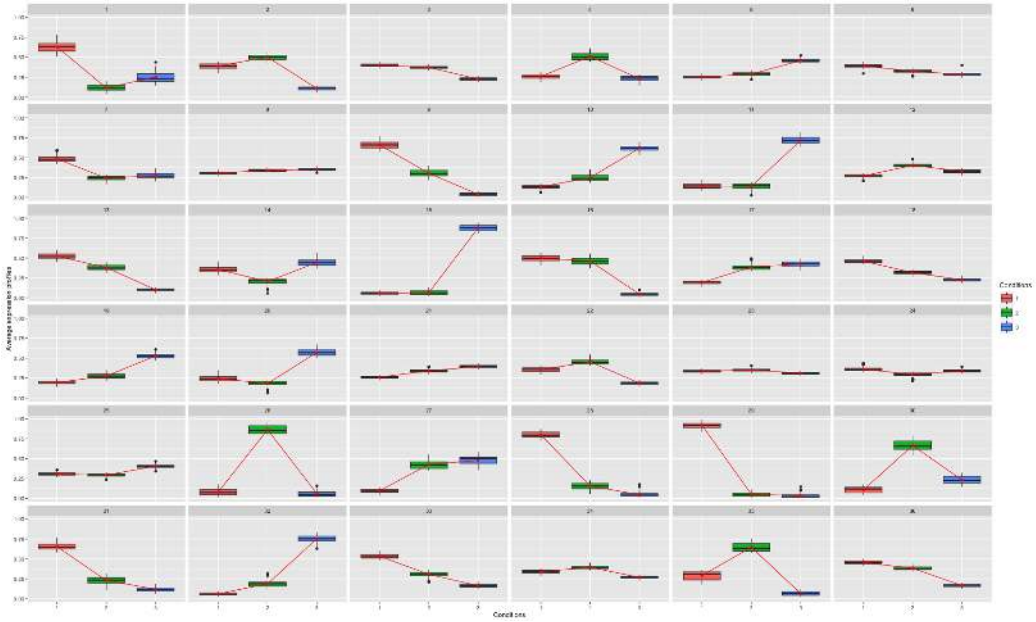

Figure 9: Visualization of boxplots of gene profiles in each cluster (average values within each condition identified by conds).

4 Further reading

For additional information on the statistical method illustrated in this vignette, see [3] and [2].

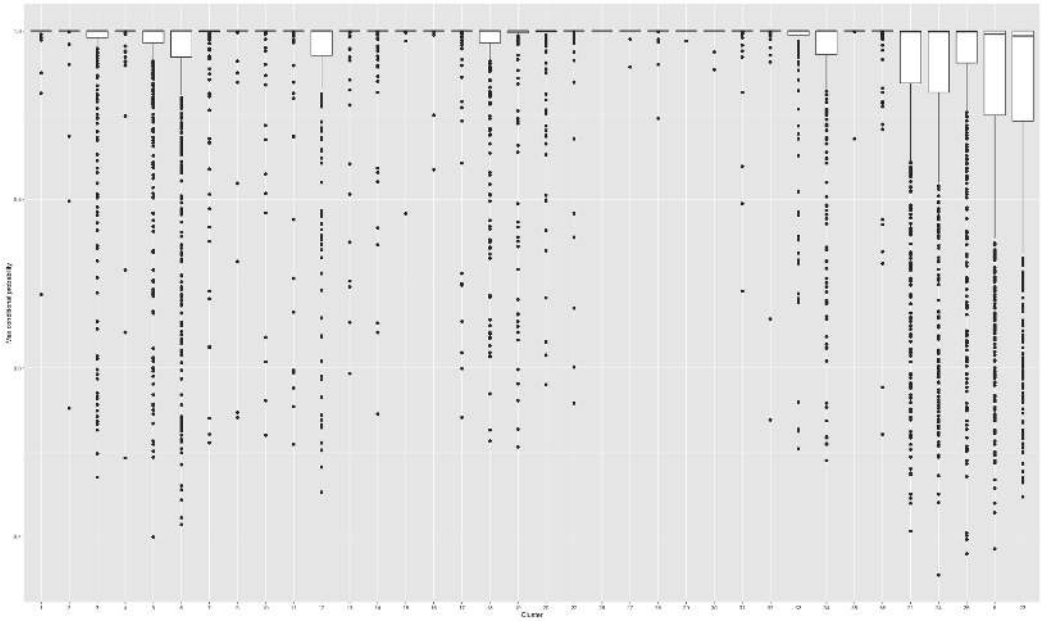

Figure 10: Boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster.

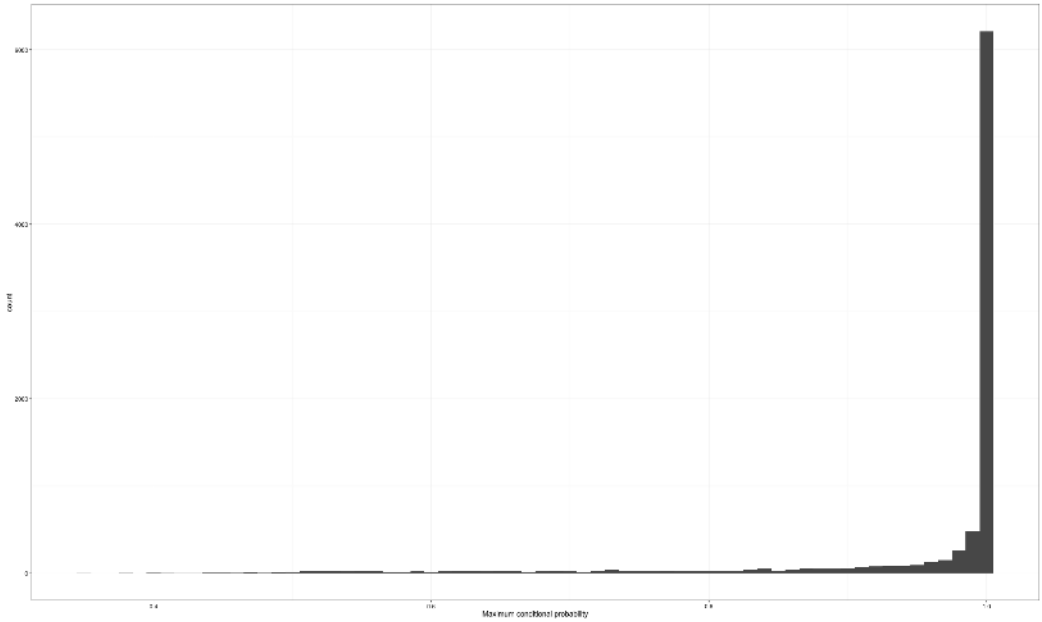

Figure 11: Histogram of maximum conditional probabilities of cluster membership.

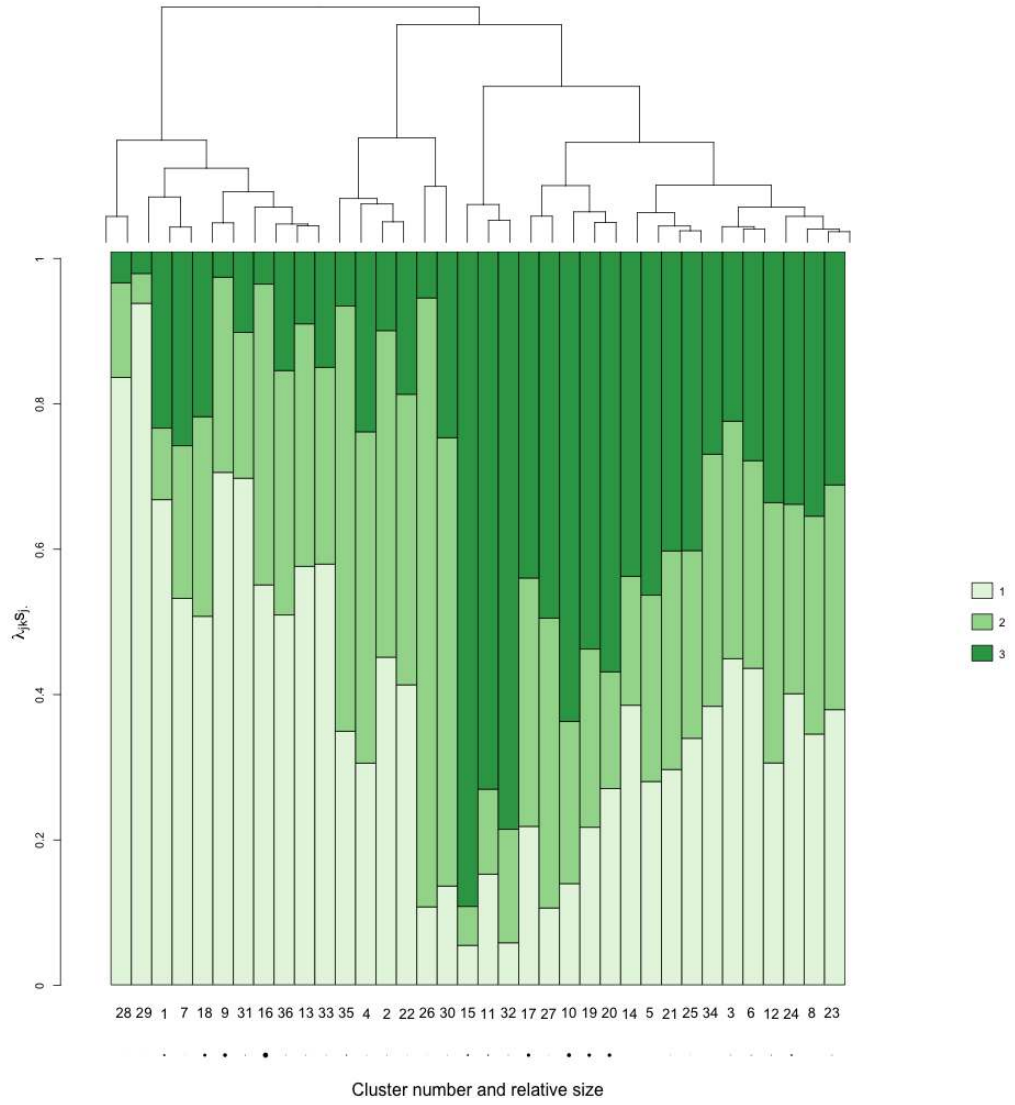

Figure 12: Visualization of overall cluster behavior for the mouse embryonic neocortex data. For each cluster, bar plots of $\hat{\lambda}_{jk}\hat{s}_{j\cdot}$ are drawn for each experimental condition, where the width of each bar corresponds to the estimated proportion $\hat{\pi}_k$

References

[1] Simone A. Fietz, Robert Lachmann, Holger Brandl, Martin Kircher, Nikolay Samusik, Roland Schröder, Naharajan Lakshmanaperumal, Ian Henry, Johannes Vogt, Axel Riehn, Wolfgang Distler, Robert Nitsch, Wolfgang Enard, Svante Pääbo, and Wieland B. Huttner. Transcriptomes of germinal zones of human and mouse fetal neocortex suggest a role of extracellular matrix in progenitor self-renewal. *PNAS*, 109(29):11836–11841, 2012.

[2] A. Rau and C. Maugis-Rabusseau. Transformation and model choice for rna-seq co-expression analysis. Technical report, ??, 2016.

[3] A. Rau et al. Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. *Bioinformatics*, 2015.

[4] Mark Ziemann, Antony Kaspi, Ross Lazarus, and Assam El-Osta. Digital Expression Explorer: A user-friendly repository of uniformly processed RNA-seq data. In *ComBio2015*, volume POS-TUE-099, Melbourne, 2015.

[5] J.-P. Baudry et al. Slope heuristics: overview and implementation. *Stat. Comp.*, 22:455–470, 2012.

[6] M.D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(R25), 2010.