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

Andrea Rau[1], Cathy Maugis-Rabusseau, Marie-Laure Martin-Magniette, and Gilles Celeux

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

HTSCluster version 2.0.4

Abstract

This vignette illustrates the use of the *HTSCluster* package through a toy example of a co-expression analysis using the human RNA-seq data from Sultan *et al.* (2008) [1]. For a full presentation of the statistical method, please see our paper [2].

Contents

1 Input data

In this vignette, we will work with the gene-level read counts from the Sultan *et al.* (2008) data [1], which may be found in the *HTSFilter* Bioconductor package [3]. These data were obtained from a human embryonic kidney (HEK293T) and a Ramos B cell line, with two biological replicates in each experimental condition. The raw read counts for 9010 genes and phenotype tables were originally obtained from the ReCount online resource [4].

We begin by loading the necessary packages, data, and phenotypic information for the analysis.

```
> library(HTSCluster)
> library(HTSFilter)
> library(Biobase)
> data(sultan)
> conds <- as.vector(phenoData(sultan)$cell.line)
> y <- exprs(sultan)
```

As an additional pre-processing step, we apply the data-based filter proposed in the `HTSFilter` package [3] to remove weakly expressed genes across the two conditions. This approach identifies a filtering threshold by maximizing a global Jaccard similarity index calculated between replicates within each condition. After applying this threshold (see Figure 1), 4956 genes were retained for the subsequent co-expression analysis.

```
> y.filter <- HTSFilter(y, conds, norm="TMM")
> table(y.filter$on)  ## 4054 off, 4956 on
```

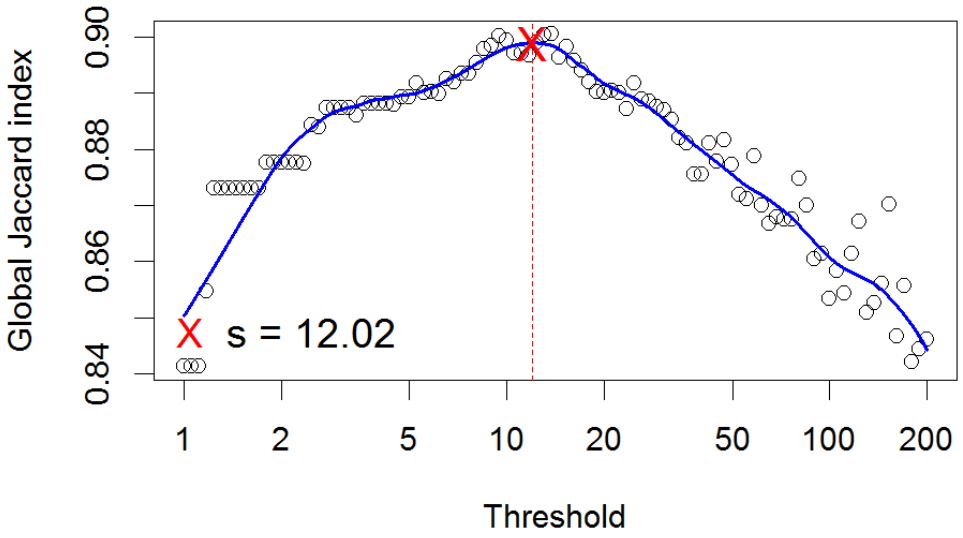

Figure 1: Global Jaccard similarity index calculated over various filtering thresholds for normalized counts for the [1] data via the `HTSFilter` package [3]. The data-driven filtering threshold in this case is equal to $s^\star = 12.02$.

```
   0    1
4054 4956

> dat.select <- y.filter$filteredData
```

2 Identifying co-expressed genes

2.1 Model description

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

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.

2.1.1 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.

2.1.2 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}.

2.1.3 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).

2.2 Co-expression analysis of Sultan *et al.* (2008) data

We perform a single run of HTSCluster for $K = 1, \ldots, 35$ clusters, using the Trimmed Means of M-values (TMM) normalization [6], and the splitting small-EM strategy described in the main paper. In the interest of reduced computational time, the settings here differ slightly from those used in the full analysis (i.e., smaller set of models and a single run); as such, the results presented here differ slightly from those presented in the main paper [2].

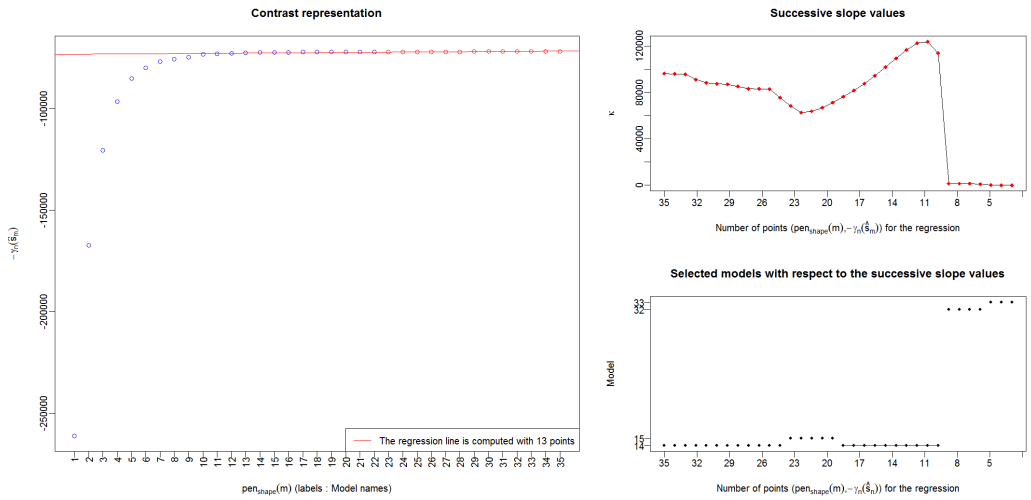

Figure 2: Diagnostic plots provided by *capushe* package for the DDSE approach; see [5] for additional information.

```
> ## ATTENTION: this code is somewhat long to run
> set.seed(12345)
> PMM <- PoisMixClusWrapper(y=dat.select, gmin=1, gmax=35,
+     conds=conds, split.init=TRUE, lib.type="TMM")
```

The above code takes about 20 minutes of computation time on a Dell Latitude E6530 quad-core 2.70 GHz Intel(R) Core(TM) with 10GB of RAM, running a 64-bit version of Windows 7 Professional.

2.2.1 Model selection

In *HTSCluster*, model selection may be performed using the DDSE calibration for slope heuristics, Djump calibration for slope heuristics, Bayesian Information Criterion (BIC), and Integrated Completed Likelihood (ICL) criterion. The models selected with each of these approaches may be accessed as follows:

```
> mod.BIC <- PMM$BIC.results
> mod.ICL <- PMM$ICL.results
> mod.Djump <- PMM$Djump.results
> mod.DDSE <- PMM$DDSE.results
```

The number of clusters selected by each of these model selection approaches may be viewed via a summary function called on the `PoisMixClusWrapper` object:

```
> summary(PMM)

**********************************************
Selected number of clusters via ICL = 10
Selected number of clusters via BIC = 30
Selected number of clusters via Djump = 15
Selected number of clusters via DDSE = 14
**********************************************
```

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.

Results from the *capushe* package over the set of models fit by *HTSCluster* may be found in the `capushe` subset of objects of class `HTSClusterWrapper` (i.e., the output of the `PoisMixClusWrapper` function). To access the results, diagnostic plots (see Figure 2), and the selected model dimension of the DDSE method, the following code may be used.

```
> DDSE <- PMM$capushe@DDSE          ## DDSE results
> plot(DDSE, newwindow=F, ask=F)    ## DDSE diagnostic plots
> DDSE@model                        ## Model selected by DDSE
```

```
[1] "14"
```

Also, note that all *capushe* diagnostic plots may be obtained directly from the `HTSClusterWrapper` object using the following command:

```
> ## Not run:
> ## plot(PMM, graphs="capushe")
```

Finally, a warning message is produced by *capushe* if the models returned by the DDSE and Djump slope heuristics approaches are not the same.

2.2.2 Visualizing results

For the following summarization and visualization, we will make use of the model selected by the DDSE approach:

```
> mod <- PMM$DDSE.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 $\hat{\boldsymbol{\pi}}$.

```
> summary(mod)
```

```
**********************************************
Number of clusters = 14
Model selection via DDSE
**********************************************
Cluster sizes:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
       540        192        235         81        458         99        514
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
       207        492        214        396        442        575        511

Number of observations with MAP > 0.90 (% of total):
1735 (35%)

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
 479       145       152       44        110       40        243
 (88.7%)   (75.52%)  (64.68%)  (54.32%)  (24.02%)  (40.4%)   (47.28%)
 Cluster 8 Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
 85        85        68         78         51         69         86
 (41.06%)  (17.28%)  (31.78%)   (19.7%)    (11.54%)   (12%)      (16.83%)

Lambda:
             Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5 Cluster 6
HEK293T           1.77      0.01      1.64      0.09      1.23      0.24
Ramos B cell      0.01      2.27      0.18      2.16      0.70      1.98
             Cluster 7 Cluster 8 Cluster 9 Cluster 10 Cluster 11 Cluster 12
HEK293T           1.39      0.41      1.13       0.59       0.72       1.02
Ramos B cell      0.50      1.76      0.84       1.52       1.36       0.97
             Cluster 13 Cluster 14
HEK293T            0.94       0.83
Ramos B cell       1.07       1.21

Pi:
 Cluster 1  Cluster 2  Cluster 3  Cluster 4  Cluster 5  Cluster 6  Cluster 7
```

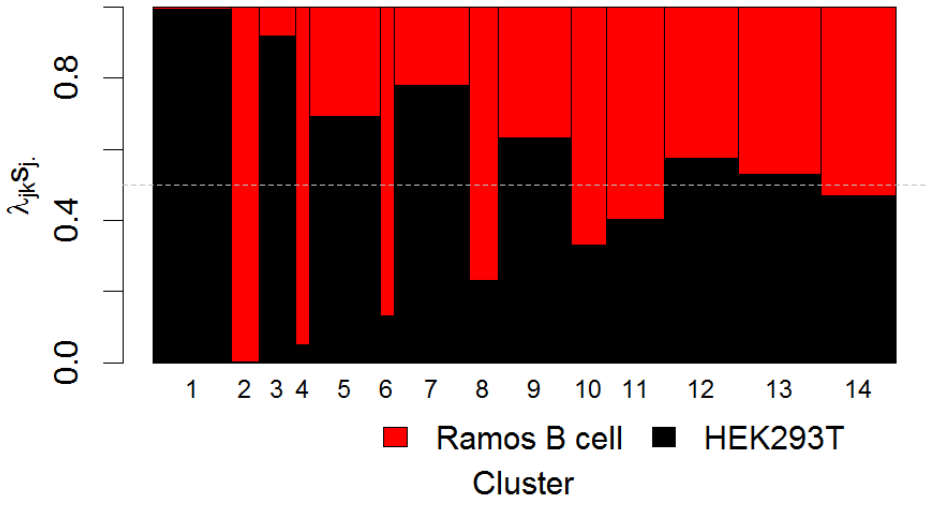

Figure 3: Visualization of overall cluster behavior for the Sultan *et al.* data. For each cluster, bar plots of $\hat{\lambda}_{jk}s_{j.}$ are drawn for each experimental condition, where the width of each bar corresponds to the estimated proportion $\hat{\pi}_k$

```
     0.11        0.04        0.05        0.02        0.09        0.02        0.10
 Cluster 8  Cluster 9 Cluster 10 Cluster 11 Cluster 12 Cluster 13 Cluster 14
     0.04        0.10        0.05        0.08        0.10        0.11        0.10
```

The estimated values for $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\pi}}$ may also be visualized using barplots, as in Figure 3, where bar widths represent the values of $\hat{\boldsymbol{\pi}}$.

```
> plot(mod, graphs="lambda")
```

Finally, we may also examine a histogram of maximum conditional probabilities of cluster membership for all genes (Figure 4, left), as well as boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster (Figure 4, right). 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.

```
> plot(mod, graphs="map")
```

```
> plot(mod, graphs="map.bycluster")
```

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

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

3 Further reading

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

4 Session Info

```
> sessionInfo()
```

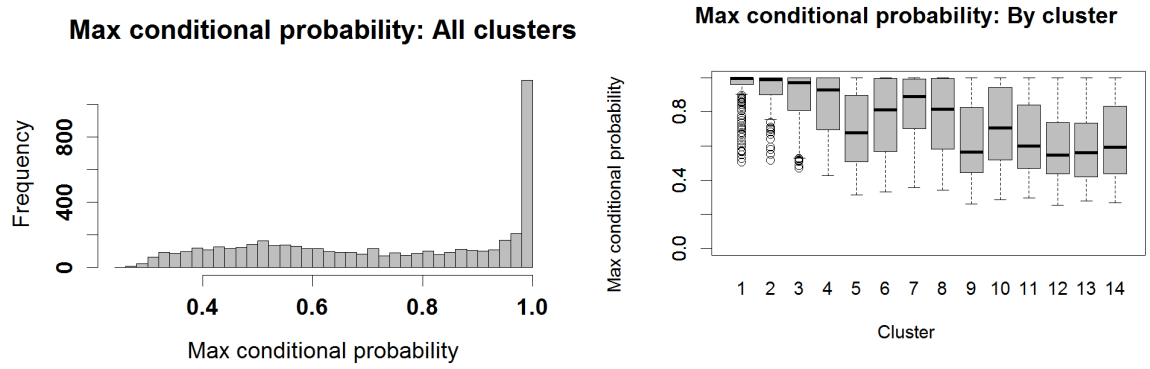

Figure 4: (left) Histogram of maximum conditional probabilities of cluster membership. (right) Boxplots of maximum conditional probabilities of cluster membership for the genes assigned to each cluster.

```
R version 3.1.1 (2014-07-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=French_France.1252  LC_CTYPE=French_France.1252
[3] LC_MONETARY=French_France.1252 LC_NUMERIC=C
[5] LC_TIME=French_France.1252

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] HTSFilter_1.4.0     Biobase_2.24.0      BiocGenerics_0.10.0
[4] HTSCluster_2.0.4    capushe_1.0         MASS_7.3-33

loaded via a namespace (and not attached):
 [1] annotate_1.42.1         AnnotationDbi_1.26.0    DBI_0.2-7
 [4] DESeq_1.16.0            DESeq2_1.4.5            edgeR_3.6.7
 [7] genefilter_1.46.1       geneplotter_1.42.0      GenomeInfoDb_1.0.2
[10] GenomicRanges_1.16.4    grid_3.1.1              IRanges_1.22.10
[13] lattice_0.20-29         limma_3.20.8            locfit_1.5-9.1
[16] plotrix_3.5-7           poisson.glm.mix_1.2     RColorBrewer_1.0-5
[19] Rcpp_0.11.2             RcppArmadillo_0.4.320.0 RSQLite_0.11.4
[22] splines_3.1.1           stats4_3.1.1            survival_2.37-7
[25] tools_3.1.1             XML_3.98-1.1            xtable_1.7-3
[28] XVector_0.4.0
```

References

[1] M. Sultan et al. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 15(5891):956–60, 2008.

[2] A. Rau et al. Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. *(submitted)*, 2014.

[3] A. Rau et al. Data-based filtering for replicated high-throughput transcriptome sequencing experiments. *Bioinf.*, 29(17):2146–2152, 2013.

[4] A. C. Frazee et al. ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC Bioinformatics*, 12(449), 2011.

[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.