Grade of Membership Model and Visualization for RNA-seq data using *CountClust*

Kushal K Dey, Chiaowen Joyce Hsiao & Matthew Stephens

Stephens Lab, The University of Chicago

*Correspondending Email: mstephens@uchicago.edu

March 10, 2016

Abstract

Grade of membership or GoM models (also known as admixture models or Latent Dirichlet Allocation") are a generalization of cluster models that allow each sample to have membership in multiple clusters. It is widely used to model ancestry of individuals in population genetics based on SNP/ microsatellite data and also in natural language processing for modeling documents [**?**, **?**].

This *R* package implements tools to visualize the clusters obtained from fitting topic models using a Structure plot [**?**] and extract the top features/genes that distinguish the clusters. In presence of known technical or batch effects, the package also allows for correction of these confounding effects.

CountClust version: 0.1.0 [1]

[1] This document used the vignette from *Bioconductor* package *DESeq2, cellTree* as *knitr* template

Contents

1 Introduction

In the context of RNA-seq expression (bulk or singlecell seq) data, the grade of membership model allows each sample to have some proportion of its RNA-seq reads coming from each cluster. For typical bulk RNA-seq experiments this assumption can be argued as follows: each sample is a mixture of different cell types, and so clusters could represent cell types (which are determined by the expression patterns of the genes), and the membership of a sample in each cluster could represent the proportions of each cell type present in that sample.

Many software packages available for document clustering are applicable to modeling RNA-seq data. Here, we use the R package `maptpx` [**?**] to fit these models, and we add functionality for visualizing the results and annotating clusters by their most distinctive genes to help biological interpretation. We also provide additional functionality to correct for batch effects and also compare the outputs from two different grade of membership model fits to the same set of samples but different in terms of feature description or model assumptions.

2 CountClust Installation

CountClust requires the following CRAN-R packages: *maptpx*, *slam*, *ggplot2*, *cowplot*,*parallel* along with the *Bioconductor* package: *limma*.

Installing *CountClust* from *Bioconductor* will install all these dependencies:

```
source("http://bioconductor.org/biocLite.R")
biocLite("CountClust")
```

For installing the working version of this package and loading the data required for this vignette, we use CRAN-R package *devtools*.

```
library(devtools)
install_github('kkdey/CountClust')
```

Then load the package with:

```
library(CountClust)
```

3 Data Preparation

We load the data as summarized experiments for the GTEx (Genotype Tissue Expression) V6 Project Brain tissue samples [**?**], the Jaitin *et al* 2014 single cell data [**?**] and the Deng *et al* 2014 single cell

data across development stages of the mouse embryo [**?**].

3.1 Jaitin et al 2014

We install the Jaitin *et al* single cell data as a summarized experiment using *devtools*.

```
library(devtools)
install_github('https://github.com/jhsiao999/singleCellRNASeqMouseJaitinSpleen.git')

install_github('kkdey/singleCellRNASeqMouseDeng2014')
```

We load the data.

```
library(singleCellRNASeqMouseJaitinSpleen)
jaitin.counts <- exprs(MouseJaitinSpleen)
jaitin.meta_data <- pData(MouseJaitinSpleen)
jaitin.gene_names <- rownames(jaitin.counts)
```

Extracting the non-ERCC genes satisfying some quality measures.

```
ENSG_genes_index <- grep("ERCC", jaitin.gene_names, invert = TRUE)
jaitin.counts_ensg <- jaitin.counts[ENSG_genes_index, ]
filter_genes <- c("M34473","abParts","M13680","Tmsb4x",
                  "S100a4","B2m","Atpase6","Rpl23","Rps18",
                  "Rpl13","Rps19","H2-Ab1","Rplp1","Rpl4",
                  "Rps26","EF437368")
fcounts <- jaitin.counts_ensg[ -match(filter_genes, rownames(jaitin.counts_ensg)), ]
sample_counts <- colSums(fcounts)

filter_sample_index <- which(jaitin.meta_data$number_of_cells == 1 &
                             jaitin.meta_data$group_name == "CD11c+" &
                                sample_counts > 600)
```

We filter the metadata likewise

```
jaitin.meta_data_filtered <- jaitin.meta_data[filter_sample_index, ]
```

3.2 Deng et al 2014

```
library(singleCellRNASeqMouseDeng2014)
deng.counts <- exprs(Deng2014MouseESC)
deng.meta_data <- pData(Deng2014MouseESC)
deng.gene_names <- rownames(counts)
```

4 Fitting GoM Model

We use a wrapper function of the *topics()* function in the *maptpx* due to Matt Taddy [**?**].

As an example, we fit the topic model for `k=7` on the Jaitin *et al* and we save the topic model output as a rda file under data folder.

```
StructureObj(t(fcounts),
            nclus_vec=7, tol=0.1,
             path_rda="../data/MouseJaitinSpleen-topicFit.rda")
```

We can also input a vector of clusters under `nclus_vec`

```
StructureObj(t(deng.counts),
            nclus_vec=2:7, tol=0.1,
             path_rda="../data/MouseDeng2014-topicFit.rda")
```

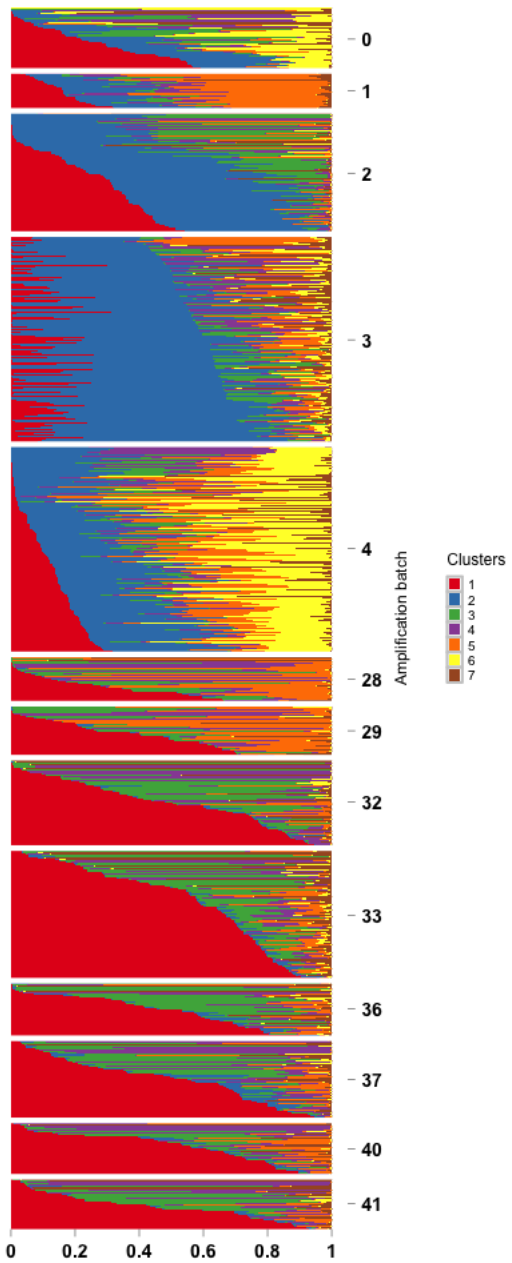
5 Structure plot visualization

Load the rda file

```
#data(MouseJaitinSpleen-topicFit)
MouseJaitinSpleen.topicFit <- readRDS("../data/MouseJaitinSpleen-topicFit.rds")
names(MouseJaitinSpleen.topicFit$clust_7)
```

```
## [1] "K"     "theta" "omega" "BF"    "D"     "X"
```

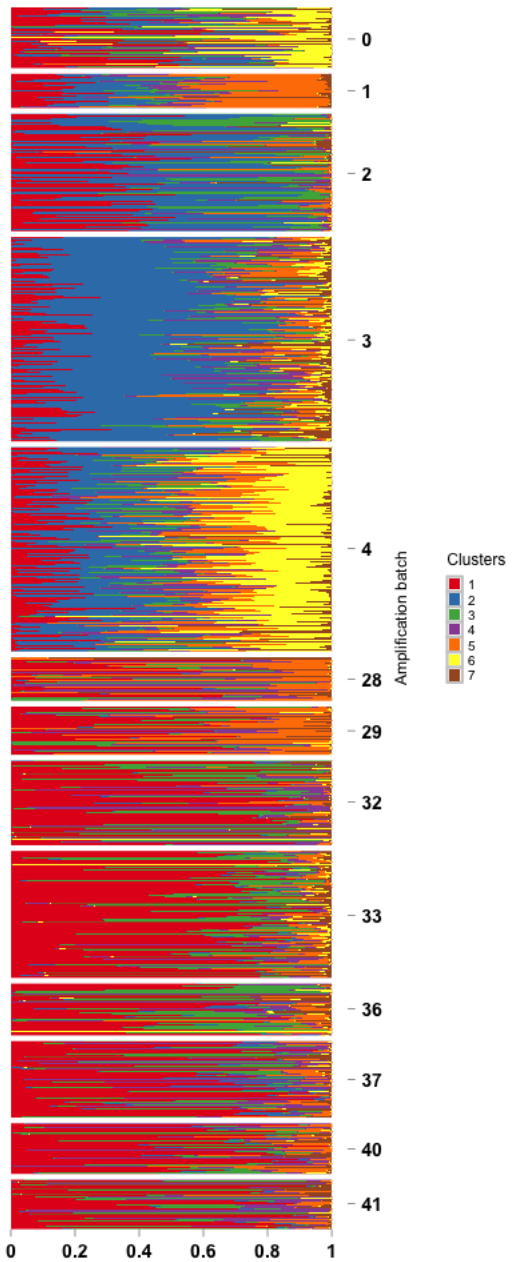
```
omega <- MouseJaitinSpleen.topicFit$clust_7$omega

amp_batch <- as.numeric(jaitin.meta_data_filtered[ , "amplification_batch"])
annotation <- data.frame(
    sample_id = paste0("X", c(1:NROW(omega)) ),
    tissue_label = factor(amp_batch,
                          levels = rev(sort(unique(amp_batch))) ) )

StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(9, "Set1"),
                yaxis_label = "Amplification batch",
                order_sample = TRUE,
                axis_tick = list(axis_ticks_length = .1,
                                 axis_ticks_lwd_y = .1,
                                 axis_ticks_lwd_x = .1,
                                 axis_label_size = 7,
                                 axis_label_face = "bold"))
```



In the above plot, the samples within each batch is sorted by the proportion of representedness of the principal cluster driving it. One can use `order_sample=FALSE` to retain the order of the samples as in the data.

```
StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(9, "Set1"),
                yaxis_label = "Amplification batch",
                order_sample = FALSE,
                axis_tick = list(axis_ticks_length = .1,
```

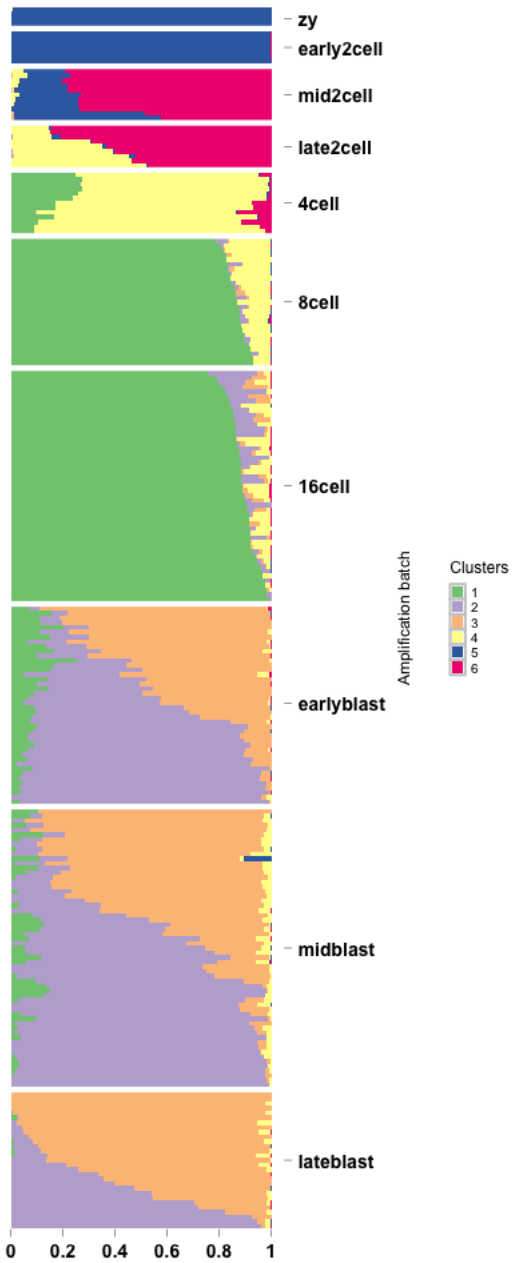
```
                                        axis_ticks_lwd_y = .1,
                                        axis_ticks_lwd_x = .1,
                                        axis_label_size = 7,
                                        axis_label_figure = "bold"))
```



Now we perform the visualization for k=6.

```
MouseDeng2014.topicFit <- get(load("../data/MouseDeng2014-topicFit.rda"))
names(MouseDeng2014.topicFit$clust_6)
```

```
## [1] "K"     "theta" "omega" "BF"    "D"     "X"
omega <- MouseDeng2014.topicFit$clust_6$omega


annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(rownames(omega),
                        levels = rev( c("zy", "early2cell", "mid2cell", "late2cell","4cell

rownames(omega) <- annotation$sample_id;

StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Amplification batch",
                order_sample = TRUE,
                axis_tick = list(axis_ticks_length = .1,
                                 axis_ticks_lwd_y = .1,
                                 axis_ticks_lwd_x = .1,
                                 axis_label_size = 7,
                                 axis_label_face = "bold"))
```



References

[1] Pritchard, Jonathan K., Matthew Stephens, and Peter Donnelly. 2000. Inference of population structure using multilocus genotype data. *Genetics*. 155.2, 945-959.

[2] Rosenberg NA, Pritchard JK, Weber JL, Cann HM, Kidd KK, Zhivotovsky LA, Feldman MW. 2002. The genetic structure of human populations. *Science*. 298, 2381-2385.

[3] Blei DM, Ng AY, Jordan MI. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*. 3, 993-1022

[4] Matt Taddy. 2012. On Estimation and Selection for Topic Models. *AISTATS 2012, JMLR W&CP 22.*(maptpx R package).

[5] Jaitin DA, Kenigsberg E et al. 2014. Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types. *Science*. 343 (6172) 776-779.

[6] Deng Q, Ramskold D, Reinius B, Sandberg R. 2014. Single-Cell RNA-Seq Reveals Dynamic, Random Monoallelic Gene Expression in Mammalian Cells. *Science*. 343 (6167) 193-196.

[7] The GTEx Consortium. 2013. The Genotype-Tissue Expression (GTEx) project. *Nature genetics*. 45(6): 580-585. doi:10.1038/ng.2653.

6 Session Info

```
sessionInfo()

## R version 3.2.2 (2015-08-14)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.5 (Yosemite)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] cowplot_0.6.0
## [2] reshape2_1.4.1
## [3] ggplot2_2.0.0
## [4] singleCellRNASeqMouseDeng2014_0.99.0
## [5] singleCellRNASeqMouseJaitinSpleen_0.99.0
## [6] Biobase_2.30.0
## [7] BiocGenerics_0.16.1
## [8] CountClust_0.1.0
## [9] knitr_1.12.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.3        digest_0.6.9       plyr_1.8.3         grid_3.2.2
##  [5] gtable_0.1.2       formatR_1.2.1      magrittr_1.5       scales_0.3.0
##  [9] evaluate_0.8       highr_0.5.1        stringi_1.0-1      labeling_0.3
## [13] BiocStyle_1.8.0    RColorBrewer_1.1-2 tools_3.2.2        stringr_1.0.0
## [17] munsell_0.4.2      colorspace_1.2-6
```