Semi supervised Grade of Membership models for RNA-seq data using *classtpx*

Kushal K Dey, Chiaowen Joyce Hsiao & Matthew Stephens

Stephens Lab, The University of Chicago

[*]Correspondending Email: mstephens@uchicago.edu

March 25, 2016

Contents

1 Introduction

The Grade of Membership (GoM) model, as fitted by the `topics()` function of the package *maptpx* or the `FitGoM()` in the package CountClust, suffers from the issue of identifiability, the reason being clusters that are determined in a completely unsupervised way. Oftentimes, we may have information about expression patterns of the biological clusters of interest. To cite an example, suppose we have RNA-seq data on blood samples (bulk or single cell) and also RNA-seq data on some FACS sorted single cells from Blood identifying different Blood cell types. Sequencing these FACS sorted cells gives us information of the expression patterns of the underlying cell types. One can view these cell types as potential clusters in clustering of RNA-seq data (non FACS sorted) from Blood. It is this prior information about the clusters that can ultimately lead to more distinct patterns of expression compared to unsupervised models and give a better sense about the mixing proportions of different cell types of interest in the samples.

2 Data Preparation

Recently a number of studies have been published on FACS sorted RNA-sequqnicng data with the aim of identifying distinct cell types of cell cycle phases. We are trying to build a library of data packages that seem relevant to our research interest and have them available as ExpressionSet objects (integrating the expression data succinctly with metadata information on samples and features). You can find a number of these data packages hosted on our Github pages https://github.com/kkdey?tab=repositories and https://github.com/jhsiao999?tab=repositories.

We present an example of the Leng et al 2015 data [paper site: http://www.nature.com/nmeth/journal/v12/n10/full/nmeth.3549.html] of RNA-sequencing data on human embryonic stem cells. Total 213 H1 single cells and 247 H1-Fucci single cells were sequenced. The 213 H1 cells were used to evaluate Oscope in identifying oscillatory genes. The H1-Fucci cells were used to confirm the cell cycle gene cluster identified by Oscope in the H1 hESCs. In the dataset, we had cells labeled H1 (213), G1 (91), S (80) and G2 (76).

```
devtools::install_github("kkdey/singleCellRNASeqHumanLengESC", force=TRUE)
```

```
library(singleCellRNASeqHumanLengESC)
data("HumanLengESC")
leng_gene_names <- Biobase::featureNames(HumanLengESC);

leng_data <- t(Biobase::exprs(HumanLengESC));
leng_metadata <- Biobase::pData(HumanLengESC)
leng_cell_state <- leng_metadata$cell_state;

table(leng_cell_state)

## leng_cell_state
##  G1  G2  H1   S
##  91  76 213  80
```

3 Methods and Materials

The general framework for Grade of Membership (GoM) models is as follows.

suppose c_{ng} represents the read counts for sample n and gene g. Then we assume the model

$$(c_{n1}, c_{n2}, \cdots, c_{nG}) \sim Mult\left(c_{n+}, p_{n1}, p_{n2}, \cdots, p_{nG}\right) \tag{1}$$

where p_{ng} represents the probability of observing a read mapping to gene g from sample n. We write this probability as

$$p_{ng} = \sum_{k=1}^{K} \omega_{nk}\theta_{kg} \qquad \sum_{k=1}^{K} \omega_{nk} = 1 \quad \forall n \qquad \sum_{g=1}^{G} \theta_{kg} = 1 \quad \forall k \tag{2}$$

where ω_{nk} represents membership probability of k th cluster in the n th sample and θ_{kg} represents cluster mass at gene g for cluster k.

In standard GoM models, the priors on ω and θ are non-informative.

$$(\omega_{n1}, \omega_{n2}, \cdots, \omega_{nK}) \sim Dir\left(\frac{1}{K}, \frac{1}{K}, \cdots, \frac{1}{K}\right) \tag{3}$$

$$(\theta_{k1}, \theta_{k2}, \cdots, \theta_{kG}) \sim Dir\left(\frac{1}{KG}, \frac{1}{KG}, \cdots, \frac{1}{KG}\right) \tag{4}$$

Now in *classtpx*, we assume that for some samples, the class labels or cluster labels are known. This information is used to either drive the θ matrix or the ω matrix. For instance, in the Leng et al 2015 data, the three classes may be considered to be the G1, S and G2 phases. There are three methods we propose

- **omega.fix**: We fix the ω vector for the samples for which the class labels are known. For instance, in the Leng et al 2015 data, for a classtpx model with K=3 representing the clusters due to G1, S and G2 phases, if the sample n comes from the G1 phase, we fix $\omega_{n.} = (1,0,0)$. Similarly, if the sample comes from S or G2 phase, we fix $\omega_{n.}$ to be $(0,1,0)$ and $(0,0,1)$ respectively. For the cells corresponding to H1 phase, the ω vector is not known and estimated from the data. In mathematical terms, we can write the model for ω as follows

$$(\omega_{n1},\omega_{n2},\cdots,\omega_{nK}) = e_k \qquad if \quad class(n) = k$$
$$(\omega_{n1},\omega_{n2},\cdots,\omega_{nK}) \sim Dir\left(\frac{1}{K},\frac{1}{K},\cdots,\frac{1}{K}\right) \qquad if \quad class(n) = NULL$$

  where $class(n)$ represents the class label of the sample (it is NULL if the class label is not known). e_k is the vector with 1 at position k and 0 at all other positions of the vector. We then perform updates on the ω on the NULL class label samples and the θ matrix and the updating scheme is similar to the `topics()` as in *maptpx* package due to Matt Taddy.
- **theta.prior**: For each class label k, we pool all samples n with $class(n) = k$, and then normalize the counts data to determine the prior θ matrix.

$$\theta_{kg} = \frac{\sum_{n:class(n)=k} c_{ng}}{\sum_g \sum_{n:class(n)=k} c_{ng}} \quad \forall g, \quad if \quad card\{n : class(n) = k\} \neq 0$$
$$(\theta_{k1},\theta_{k2},\cdots,\theta_{kG}) \sim Dir\left(\frac{1}{KG},\frac{1}{KG},\cdots,\frac{1}{KG}\right) \quad if \quad card\{n : class(n) = k\} = 0 \qquad (5)$$

  One can also apply adaptive shrinkage on the $\theta_{k.}$ values. In that case, define

$$\beta_{kg} = \frac{\sum_{n:class(n)=k} c_{ng}}{N_k} - \frac{\sum_g \sum_{n:class(n)=k} c_{ng}}{N} \qquad card\{n : class(n) = k\} = N_k \qquad (6)$$

  We assume

$$s_{kg} = \frac{1}{N_k(N_k-1)} \sum_{class(n)=k} \left(c_{ng} - \frac{\sum_{n:class(n)=k} c_{ng}}{N_k} \right)^2 \qquad card\{n : class(n) = k\} = N_k \qquad (7)$$

  Then we perform `ash` on the vector (β_{kg}, s_{kg}) over all genes g for each k and then obtain the posterior mean of β, say $\beta_{post}(kg)$.
  We then fix the θ values as

$$\theta^{\star}_{kg} = \frac{\sum_{g} \sum_{n:class(n)=k} c_{ng}}{N} + \beta_{post}(kg) \tag{8}$$

So, in other words we assume

$$\begin{aligned} \theta_{kg} &= \theta^{\star}_{kg} \quad \forall g, \quad if \quad card\{n : class(n) = k\} \neq 0 \\ (\theta_{k1}, \theta_{k2}, \cdots, \theta_{kG}) &\sim Dir\left(\frac{1}{KG}, \frac{1}{KG}, \cdots, \frac{1}{KG}\right) \quad if \quad card\{n : class(n) = k\} = 0 \end{aligned} \tag{9}$$

These $\theta_{k.}$ as in Eqn (5) or Eqn (9) (depending on whether we use shrinkage or not) are then input into the GoM model framework as prior cluster probability vectors and we update them based on the data to find the posterior estimates at each stage of iteration (the updating scheme similar to the `topics()` as in *maptpx* package due to Matt Taddy).

- **theta.fix**: In this method, instead of setting $\theta_{k.}$ matrix for k with $N_k \neq 0$ as prior, we fix them at these values and do not update them during the iterative steps. The only updates correspond to those k for which $N_k = 0$, and the updating scheme is similar to the `topics()` as in *maptpx* package due to Matt Taddy.

4 Results

We first cite the example of the Leng et al data and fit `classtpx()` model on the data.

4.1 Leng et al (2015)

We fix the class labels and the sample indices as follows.

```
index_1 <- which(leng_cell_state=="G1");
index_2 <- which(leng_cell_state=="S");
index_3 <- which(leng_cell_state=="G2");

known_samples <- c(index_1, index_2, index_3);
class_labs <- c(rep("G1", length(index_1)),
                rep("S", length(index_2)),
                rep("G2", length(index_3)));
```

Then we can perform `classtpx()` model for `omega.fix()` method as follows

```
Topic_clus <- classtpx::class_topics(
    leng_data,
    K=3,
    known_samples = known_samples,
    class_labs = class_labs,
    method="omega.fix",
    tol=0.01)

save(Topic_clus, file="../data/leng_topic_fit_3_classtpx_omega_fix.rda")
```

We can perform Structure plot visualization of the results.

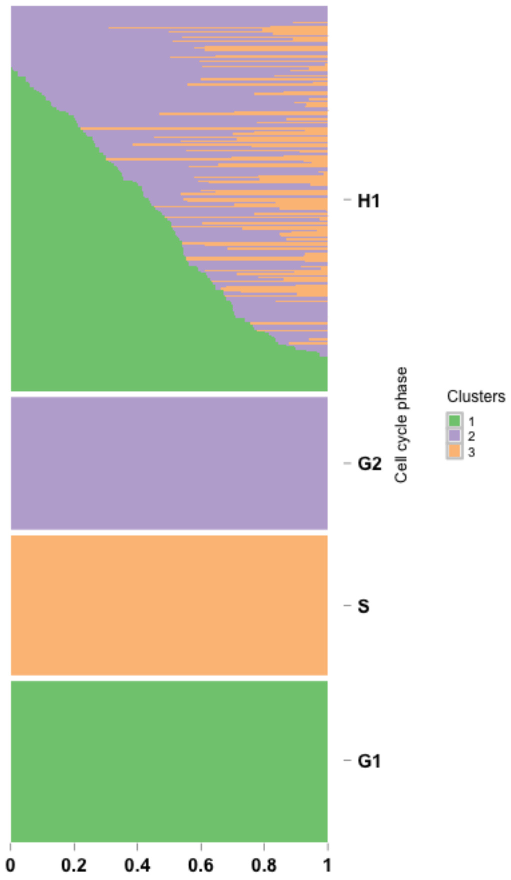
```
Topic_clus <- get(load(file="../data/leng_topic_fit_3_classtpx_omega_fix.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(leng_cell_state,
                        levels = c("G1", "S", "G2", "H1") ) )


rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



We now perform `classtpx()` model for `theta.prior()` method.

```
Topic_clus <- classtpx::class_topics(
    leng_data,
    K=3,
    known_samples = known_samples,
    class_labs = class_labs,
    method="theta.prior",
    tol=0.01,
    shrink=TRUE)

save(Topic_clus, file="../data/leng_topic_fit_3_classtpx_theta_prior.rda")
```

We can perform Structure plot visualization of the results.

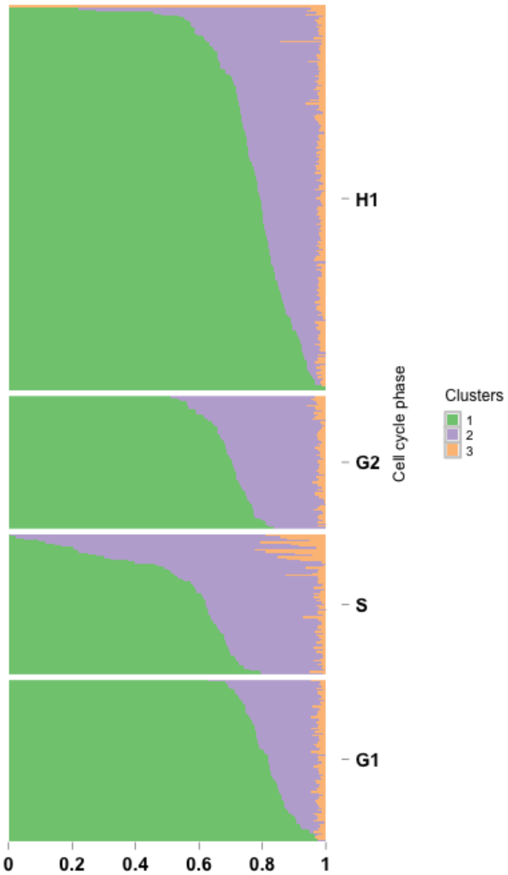
```
Topic_clus <- get(load(file="../data/leng_topic_fit_3_classtpx_theta_prior.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(leng_cell_state,
                        levels = c("G1", "S", "G2", "H1") ) )


rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



Finally we apply the `theta.fix()` method.

```
Topic_clus <- classtpx::class_topics(
    leng_data,
    K=3,
    known_samples = known_samples,
    class_labs = class_labs,
    method="theta.fix",
    tol=0.01,
    shrink=FALSE)

save(Topic_clus, file="../data/leng_topic_fit_3_classtpx_theta_fix.rda")
```

We can perform Structure plot visualization of the results.

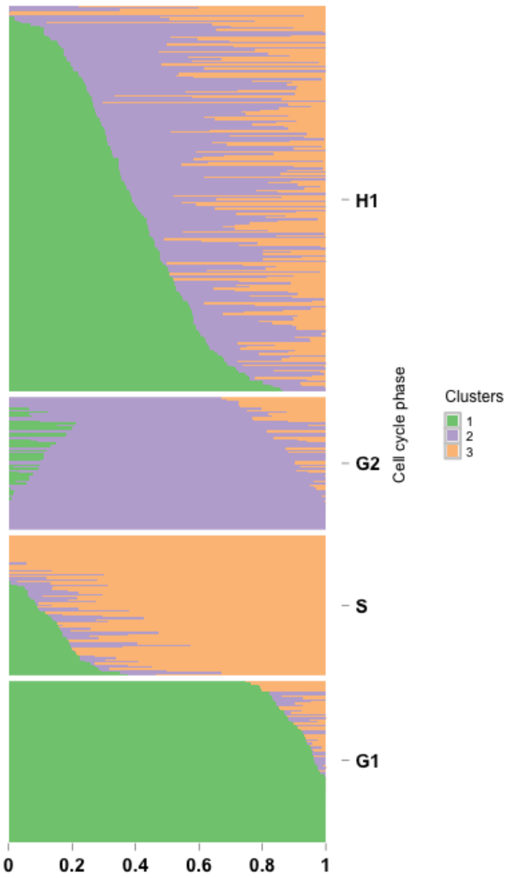
```
Topic_clus <- get(load(file="../data/leng_topic_fit_3_classtpx_theta_fix.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(leng_cell_state,
                        levels = c("G1", "S", "G2", "H1") ) )


rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



4.2 Treutlin et al (2014)

Treutlin et al 2014 sequenced single cell transcriptome data from mouse lung epithelium. The cells were collected at various stages E14.5, E16.5, E18.5 and some adult replicates. We performed both `maptpx()` model and `classtpx` model fitting on this data.

```
devtools::install_github("jhsiao999/singleCellRNASeqMouseTreutleinLung", force=TRUE)
```

```
library(singleCellRNASeqMouseTreutleinLung)
data("MouseTreutleinLung")
leng_gene_names <- Biobase::featureNames(HumanLengESC);

counts_data <- t(Biobase::exprs(MouseTreutleinLung));
pheno_metadata <- pData(MouseTreutleinLung);
table(pheno_metadata[,1])

##
## adult E14.5 E16.5 E18.5
##    46    45    27    83
```

We first apply the maptpx model for $K = 3$.

```
Topic_clus <- maptpx::topics(counts_data, 3, tol=0.1);
save(Topic_clus, file="../data/treutlin_topic_fit_3_maptpx.rda")
```

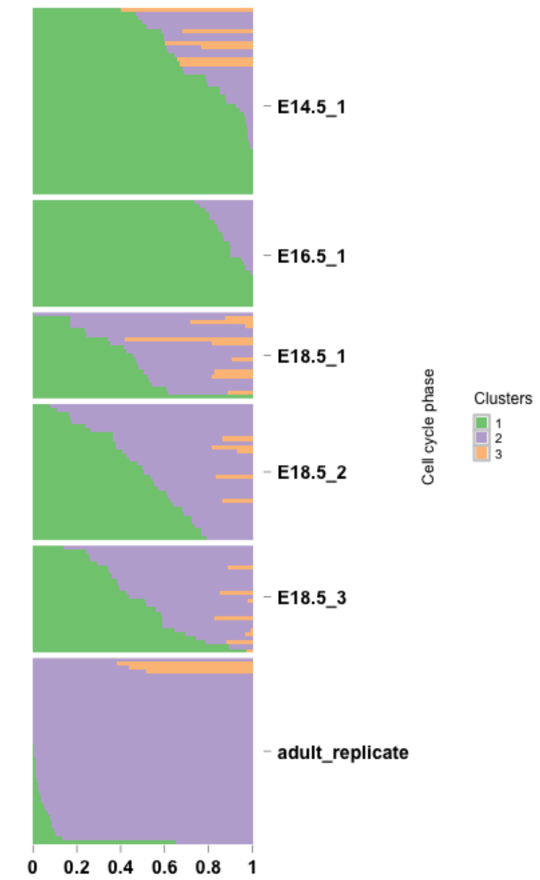
```
Topic_clus <- get(load(file="../data/treutlin_topic_fit_3_maptpx.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(paste0(pheno_metadata$cell_type, "_",
                               pheno_metadata$replicate),
                        levels=rev(c("E14.5_1", "E16.5_1",
                                     "E18.5_1","E18.5_2",
                                     "E18.5_3","adult_replicate"))
))

rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



We next performed classtpx model for K=2 with `omega.fix()` method. We chose E14.5 as one group and adult replicates as another group in defining class labels and assumed we do not have class label information for E16.5 and E18.5 phases.

```
known_samples <- c(which(pheno_metadata$cell_type=="E14.5"),
                   which(pheno_metadata$cell_type=="adult"));
class_labs <- c(rep(1, length(which(pheno_metadata$cell_type=="E14.5"))),
                rep(2,length(which(pheno_metadata$cell_type=="adult"))));
```

```
Topic_clus <- classtpx::class_topics(
    counts_data,
    K=2,
    known_samples = known_samples,
    class_labs = class_labs,
    method="omega.fix",
    tol=0.01,
    shrink=FALSE)

save(Topic_clus, file="../rdas/treutlin_topic_fit_2_classtpx_omega_fix.rda")
```

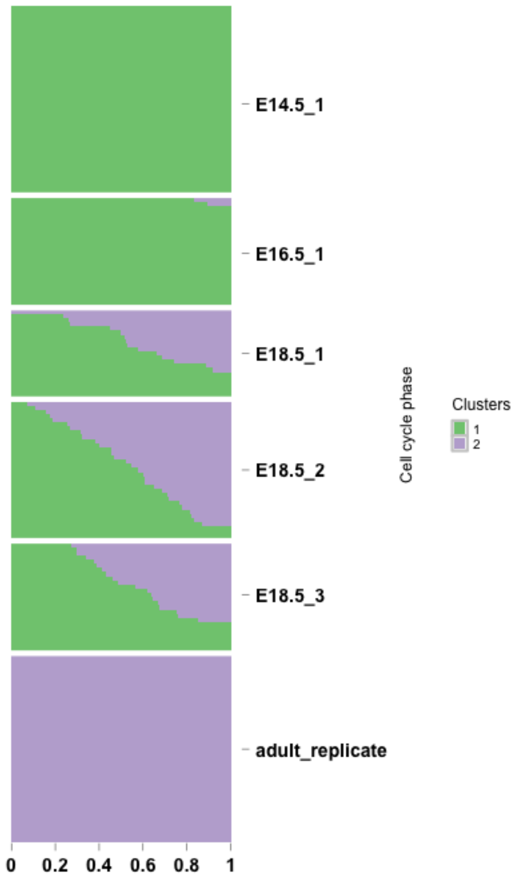
```
Topic_clus <- get(load(file="../data/treutlin_topic_fit_2_classtpx_omega_fix.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(paste0(pheno_metadata$cell_type, "_",
                               pheno_metadata$replicate),
                        levels=rev(c("E14.5_1", "E16.5_1",
                                     "E18.5_1","E18.5_2",
                                     "E18.5_3","adult_replicate"))
))

rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



We perform the method with `theta.fix()` for K=2 and K=3.

```
Topic_clus <- classtpx::class_topics(
    counts_data,
    K=2,
    known_samples = known_samples,
    class_labs = class_labs,
    method="theta.fix",
    tol=0.01,
    shrink=FALSE)

save(Topic_clus, file="../rdas/treutlin_topic_fit_2_classtpx_theta_fix.rda")
```

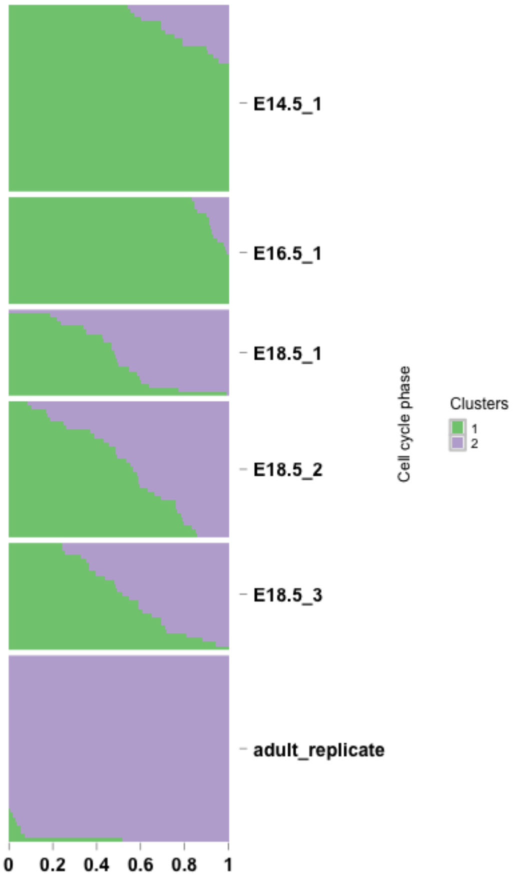
```
Topic_clus <- get(load(file="../data/treutlin_topic_fit_2_classtpx_theta_fix.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(paste0(pheno_metadata$cell_type, "_",
                               pheno_metadata$replicate),
                        levels=rev(c("E14.5_1", "E16.5_1",
                                     "E18.5_1","E18.5_2",
                                     "E18.5_3","adult_replicate"))
))

rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



```
Topic_clus <- classtpx::class_topics(
    counts_data,
    K=2,
    known_samples = known_samples,
    class_labs = class_labs,
    method="theta.fix",
    tol=0.01,
    shrink=FALSE)

save(Topic_clus, file="../rdas/treutlin_topic_fit_3_classtpx_theta_fix.rda")
```

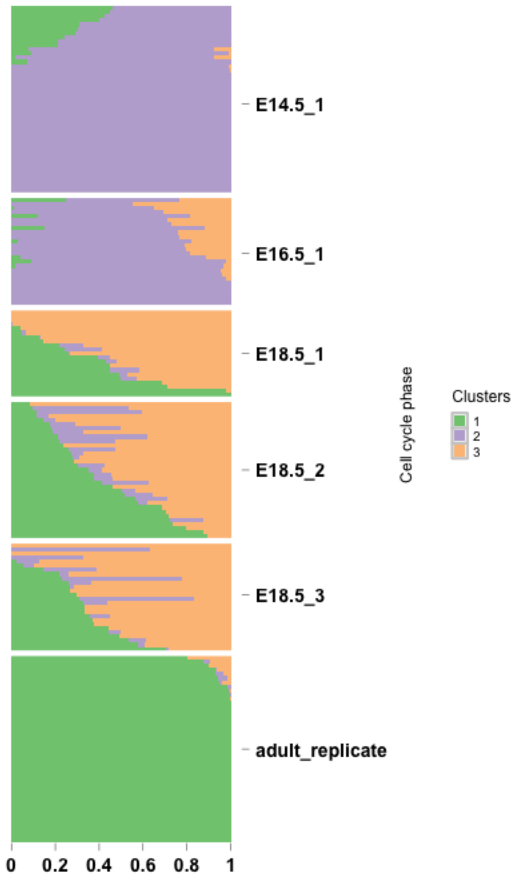
```
Topic_clus <- get(load(file="../data/treutlin_topic_fit_3_classtpx_theta_fix.rda"))

omega <- Topic_clus$omega;

annotation <- data.frame(
  sample_id = paste0("X", c(1:NROW(omega))),
  tissue_label = factor(paste0(pheno_metadata$cell_type, "_",
                               pheno_metadata$replicate),
                        levels=rev(c("E14.5_1", "E16.5_1",
                                     "E18.5_1","E18.5_2",
                                     "E18.5_3","adult_replicate"))
))

rownames(omega) <- annotation$sample_id;


CountClust::StructureGGplot(omega = omega,
                annotation = annotation,
                palette = RColorBrewer::brewer.pal(8, "Accent"),
                yaxis_label = "Cell cycle phase",
                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"))
```



5 Session Info

```
sessionInfo()

## R version 3.2.4 (2016-03-10)
## 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] singleCellRNASeqMouseTreutleinLung_0.99.0
## [2] singleCellRNASeqHumanLengESC_0.99.0
## [3] Biobase_2.30.0
## [4] BiocGenerics_0.16.1
## [5] knitr_1.12.3
##
## loaded via a namespace (and not attached):
##  [1] flexmix_2.3-13     Rcpp_0.12.3        cluster_2.0.3      magrittr_1.5
##  [5] MASS_7.3-45        munsell_0.4.3      cowplot_0.6.1      ape_3.4
##  [9] colorspace_1.2-6   lattice_0.20-33    stringr_1.0.0      highr_0.5.1
## [13] plyr_1.8.3         tools_3.2.4        nnet_7.3-12        grid_3.2.4
## [17] nlme_3.1-125       gtable_0.2.0       mgcv_1.8-12        vegan_2.3-4
## [21] maptpx_1.9-2       modeltools_0.2-21  gtools_3.5.0       digest_0.6.9
## [25] permute_0.9-0      picante_1.6-2      Matrix_1.2-4       RColorBrewer_1.1-2
## [29] reshape2_1.4.1     ggplot2_2.1.0      formatR_1.2.1      evaluate_0.8
## [33] slam_0.1-32        labeling_0.3       limma_3.26.8       stringi_1.0-1
## [37] scales_0.4.0       CountClust_0.99.3  stats4_3.2.4       BiocStyle_1.8.0
```