Trajectory Inference

Indu khatri

10/14/2019

```
knitr::opts_chunk$set(
    message = FALSE,
    warning = FALSE,
    cache = TRUE
)
```

Overview

We are going to use 2300 PBMCs single-cells from mouse to construct the trajectory. These cells are selected form 8K PBMCs from 10X website based on the high expression of Cathepsin S (CTSS) gene. This gene is lysosomal cysteine proteinase that may participate in the degradation of antigenic proteins to peptides for presentation on MHC class II molecules. Hence, it is important to track the the regulatory effect of this gene in different cells.

First, load the object.

```
setwd("~/Desktop/")
load("2K_pbmc_subset1.RData", verbose = FALSE)
```

and packages

```
suppressMessages(require(monocle))
suppressMessages(require(cellrangerRkit))
suppressMessages(require(dplyr))
```

We'll select the genes expressed in greater than 5% of the cells and order the cells based on genes.

```
my_cds_subset <- detectGenes(my_cds_subset, min_expr = 0.1)
fData(my_cds_subset)$use_for_ordering <- fData(my_cds_subset)$num_cells_expressed > 0.05 * ncol(my_cds_s
```

```
# how many genes are used?
table(fData(my_cds_subset)$use_for_ordering)
```

```
##
## FALSE  TRUE
##  8641  6805
```

We will now perform clustering but without specifying the number of clusters; we will use thresholds on the cell's local density (rho) and nearest distance (delta) to determine the number of clusters.

```
my_cds_subset <- reduceDimension(my_cds_subset,
                                 max_components = 2,
                                 norm_method = 'log',
                                 num_dim = 10,
                                 reduction_method = 'tSNE',
                                 verbose = TRUE)

my_cds_subset <- clusterCells(my_cds_subset, verbose = FALSE)
```

```
## Distance cutoff calculated to 3.679052
```

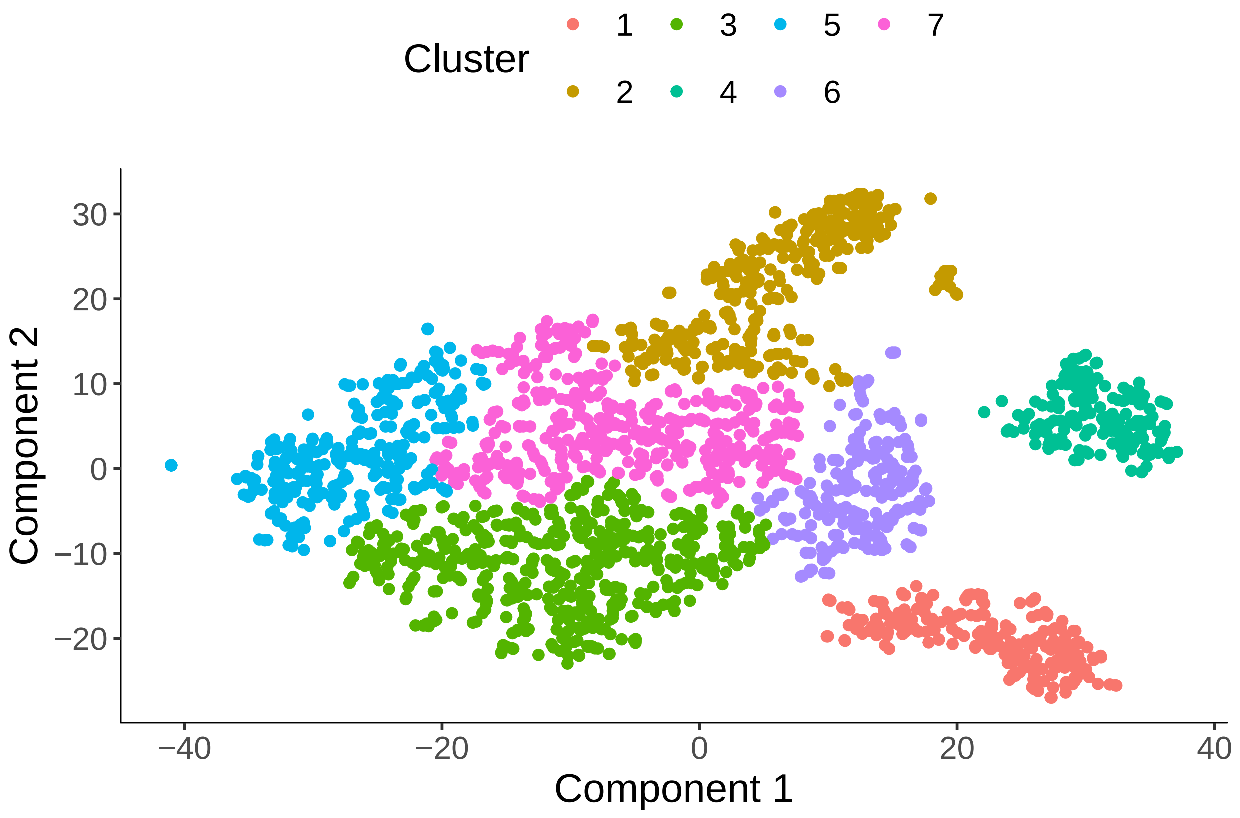
We'll use rho = 3.6 to cluster the cells again.

```r
my_cds_subset <- clusterCells(my_cds_subset,
                              rho_threshold = 3.6,
                              delta_threshold = 10,
                              skip_rho_sigma = T,
                              verbose = FALSE)
```

```r
table(pData(my_cds_subset)$Cluster)
```

```
##
##   1   2   3   4   5   6   7
## 243 356 527 195 302 231 446
```

```r
plot_cell_clusters(my_cds_subset)
```



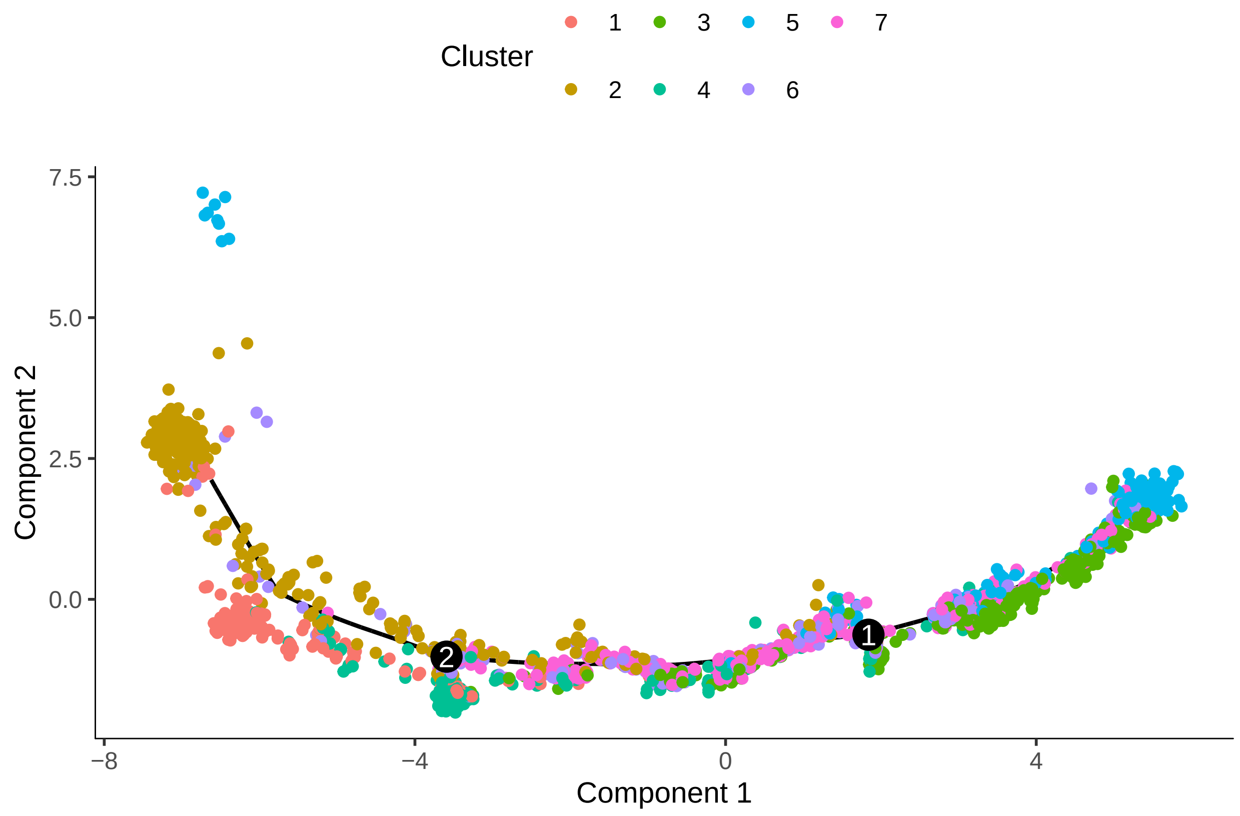
Now we'll perform the differential gene expression analysis as before but across all cell clusters.

```r
#Pay attention to number of cores. Mention only available.
clustering_DEG_genes <- differentialGeneTest(my_cds_subset,
                                             fullModelFormulaStr='~Cluster', cores = 8)
```

We'll use the top 500 most significantly differentially expressed genes as the set of ordering genes and perform the dimension reduction and the trajectory analysis (using the orderCells() function).

```r
my_ordering_genes <-row.names(clustering_DEG_genes)[order(clustering_DEG_genes$qval)][1:500]
my_cds_subset <- setOrderingFilter(my_cds_subset, ordering_genes = my_ordering_genes)
my_cds_subset <- reduceDimension(my_cds_subset, method = 'DDRTree')
my_cds_subset <- orderCells(my_cds_subset)
```

```r
plot_cell_trajectory(my_cds_subset, color_by = "Cluster")
```



Finding Genes that Change as a Function of Pseudotime

Once we have a trajectory, we can use differentialGeneTest() to find genes that have an expression pattern that varies according to pseudotime.

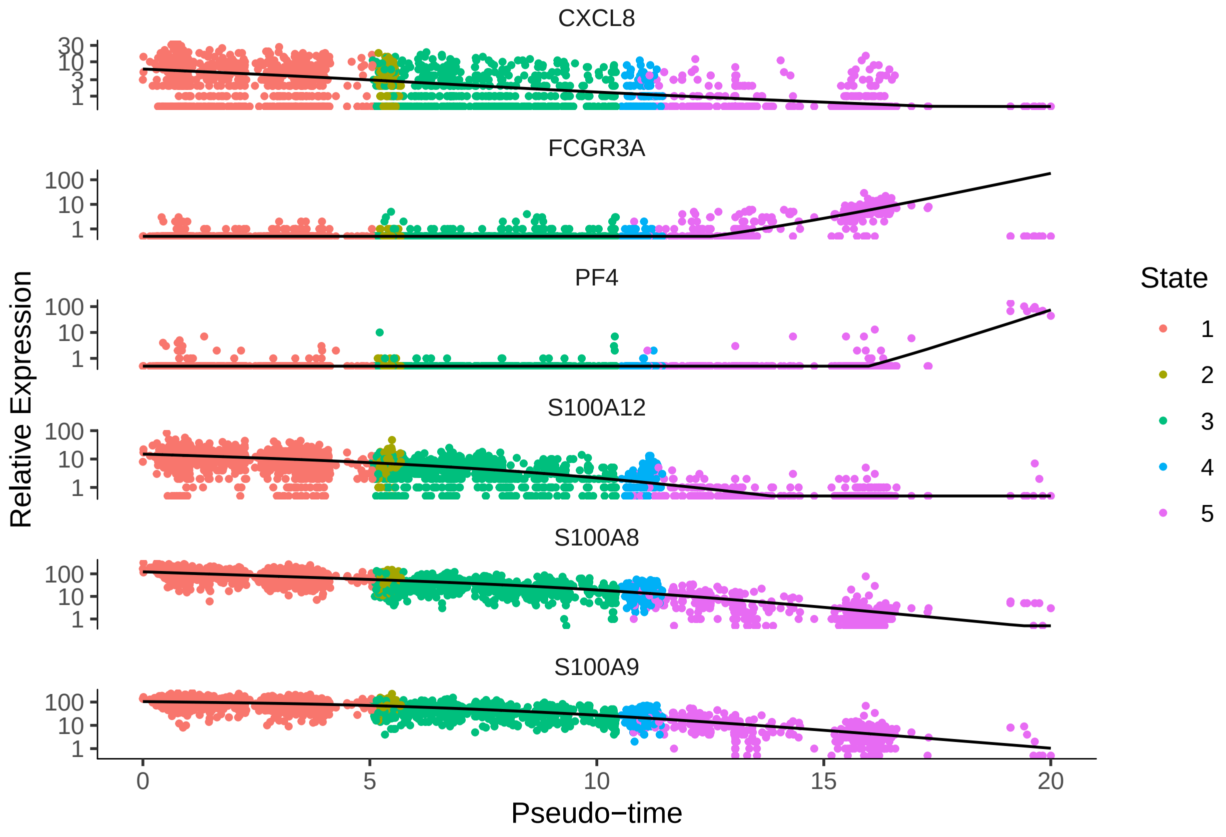
```
my_pseudotime_de <- differentialGeneTest(my_cds_subset,
                                         fullModelFormulaStr = "~sm.ns(Pseudotime)",
                                         cores = 8)

my_pseudotime_de %>% arrange(qval) %>% head()
```

```
##   status           family pval qval              id gene_short_name
## 1     OK negbinomial.size    0    0 ENSG00000163220          S100A9
## 2     OK negbinomial.size    0    0 ENSG00000163221         S100A12
## 3     OK negbinomial.size    0    0 ENSG00000143546          S100A8
## 4     OK negbinomial.size    0    0 ENSG00000203747          FCGR3A
## 5     OK negbinomial.size    0    0 ENSG00000169429           CXCL8
## 6     OK negbinomial.size    0    0 ENSG00000163737             PF4
##   num_cells_expressed use_for_ordering
## 1                2285             TRUE
## 2                1887             TRUE
## 3                2270             TRUE
## 4                 377             TRUE
## 5                1278             TRUE
## 6                  76             TRUE
```

```
my_pseudotime_de %>% arrange(qval) %>% head() %>% select(id) -> my_pseudotime_gene
my_pseudotime_gene <- my_pseudotime_gene$id
```

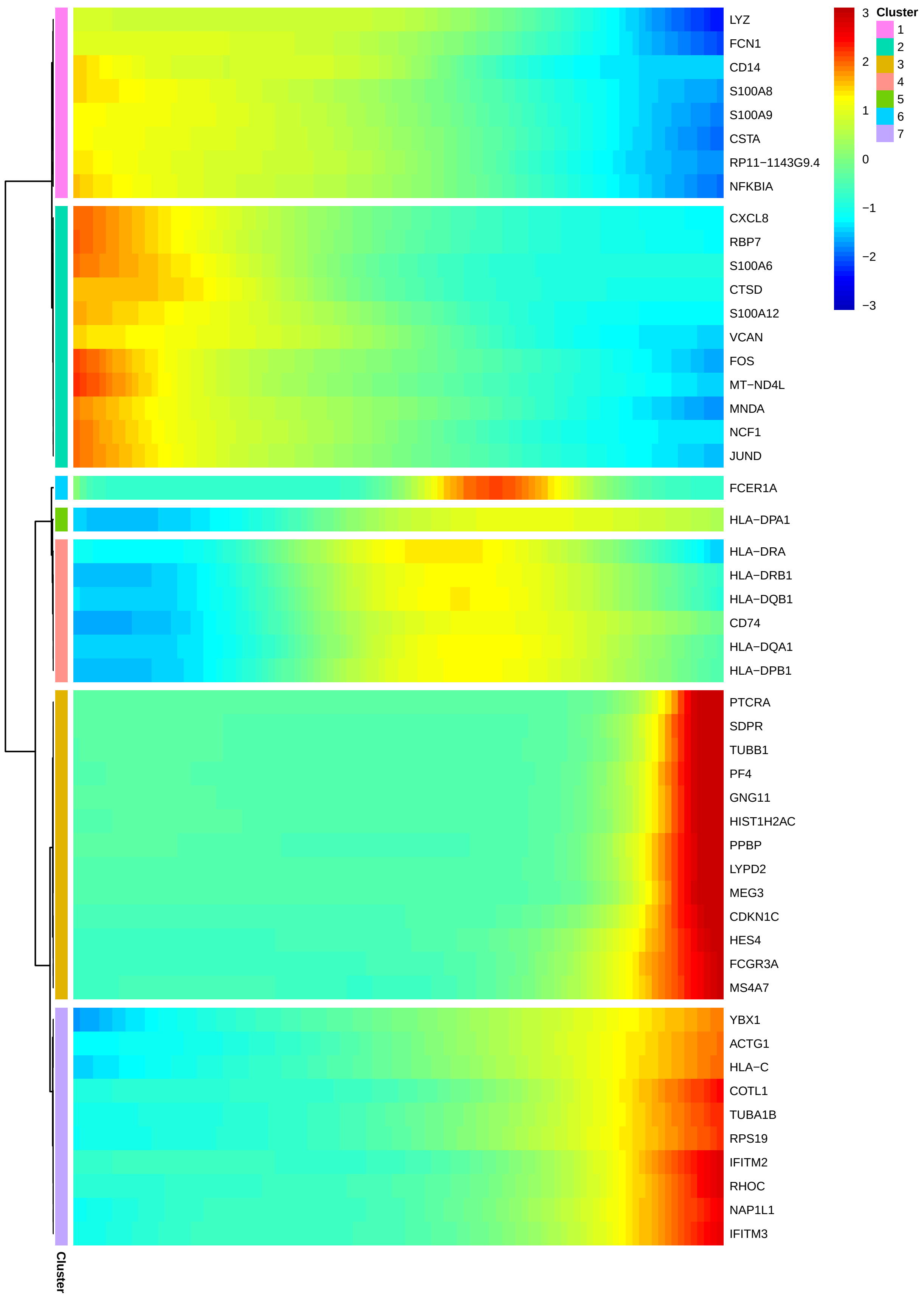
```
plot_genes_in_pseudotime(my_cds_subset[my_pseudotime_gene,])
```



Clustering Genes by Pseudotemporal Expression Pattern

```
# cluster the top 50 genes that vary as a function of pseudotime
my_pseudotime_de %>% arrange(qval) %>% head(50) %>% select(id) -> gene_to_cluster
gene_to_cluster <- gene_to_cluster$id

my_pseudotime_cluster <- plot_pseudotime_heatmap(my_cds_subset[gene_to_cluster,],
                                                 num_clusters = 7,
                                                 cores = 8,
                                                 show_rownames = TRUE,
                                                 return_heatmap = TRUE)
```

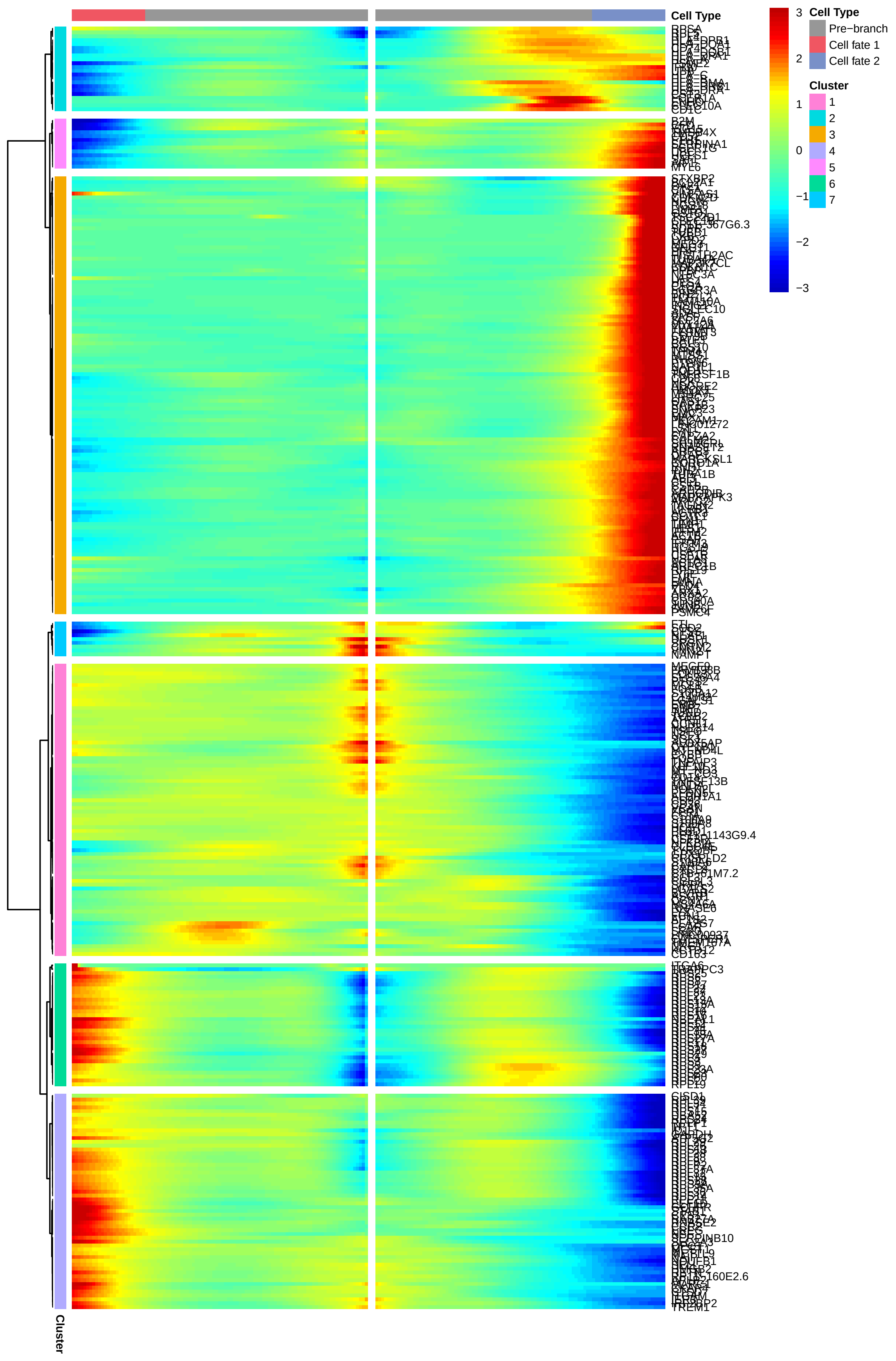

LYZ
FCN1
CD14
S100A8
S100A9
CSTA
RP11−1143G9.4
NFKBIA
CXCL8
RBP7
S100A6
CTSD
S100A12
VCAN
FOS
MT−ND4L
MNDA
NCF1
JUND
FCER1A
HLA−DPA1
HLA−DRA
HLA−DRB1
HLA−DQB1
CD74
HLA−DQA1
HLA−DPB1
PTCRA
SDPR
TUBB1
PF4
GNG11
HIST1H2AC
PPBP
LYPD2
MEG3
CDKN1C
HES4
FCGR3A
MS4A7
YBX1
ACTG1
HLA−C
COTL1
TUBA1B
RPS19
IFITM2
RHOC
NAP1L1
IFITM3
3
2
1
0
−1
−2
−3
Cluster
1
2
3
4
5
6
7
Cluster

Analyzing Branches in Single-Cell Trajectories

Our trajectory has four branches, which represents cells that have alternative gene expression patterns. These represent cells that have supposedly gone through different developmental paths. We will now identify the genes that differ at a particular branch point. Here is the trajectory again.

```r
BEAM_res <- BEAM(my_cds_subset, branch_point = 1, cores = 8)
BEAM_res <- BEAM_res[order(BEAM_res$qval),]
BEAM_res <- BEAM_res[,c("gene_short_name", "pval", "qval")]
```

```r
#The heatmap shows how some genes are over-expressed or under-expressed depending
#on the trajectory path.
my_branched_heatmap <- plot_genes_branched_heatmap(my_cds_subset[row.names(subset(BEAM_res,
                                                                      qval < 1e-4)),],
                                                   branch_point = 1,
                                                   num_clusters = 7,
                                                   cores = 8,
                                                   use_gene_short_name = TRUE,
                                                   show_rownames = TRUE,
                                                   return_heatmap = TRUE)
```


Cell Type
3
2
1
0
−1
−2
−3
Cell Type
Pre−branch
Cell fate 1
Cell fate 2
Cluster
1
2
3
4
5
6
7
Cluster

We can return genes that belong to specific clusters that were identified by BEAM().

```
head(my_branched_heatmap$annotation_row)
```

```
##                Cluster
## S100A9               1
## S100A8               1
## S100A12              1
## LYZ                  1
## RP11-1143G9.4        1
## VCAN                 1
```

```
dim(my_branched_heatmap$annotation_row)
```

```
## [1] 322   1
```

```
table(my_branched_heatmap$annotation_row$Cluster)
```

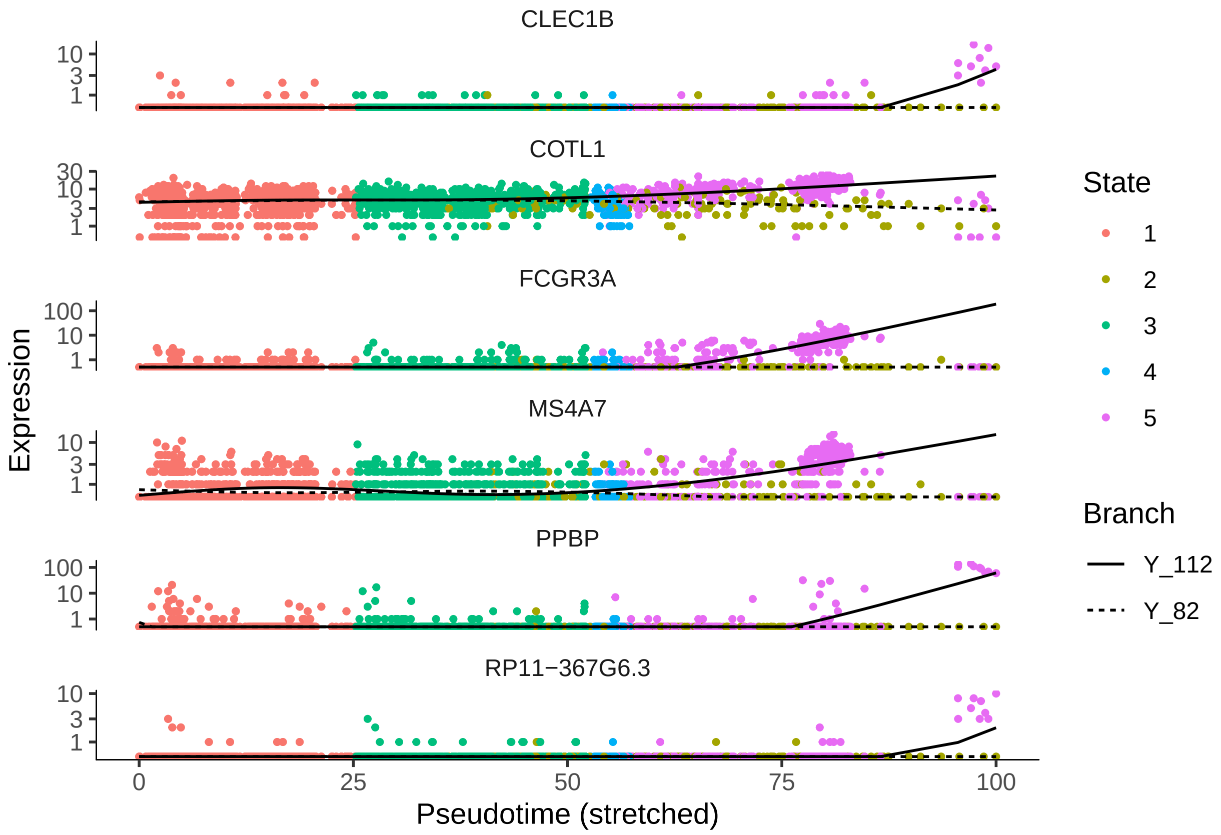
```
## 
##   1   2   3   4   5   6   7 
##  76  22 114  56  13  32   9
```

```
my_row <- my_branched_heatmap$annotation_row
my_row <- data.frame(cluster = my_row$Cluster,
                     gene = row.names(my_row),
                     stringsAsFactors = FALSE)
head(my_row[my_row$cluster == 3,'gene'])
```

```
## [1] "FCGR3A"       "CLEC1B"       "RP11-367G6.3" "COTL1"       
## [5] "PPBP"         "MS4A7"
```

```
my_gene <- row.names(subset(fData(my_cds_subset),
                            gene_short_name %in% head(my_row[my_row$cluster == 3,'gene'])))

# plot genes that are expressed in a branch dependent manner
plot_genes_branched_pseudotime(my_cds_subset[my_gene,],
                               branch_point = 1,
                               ncol = 1)
```



```r
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.14.2
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## 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] splines   stats4    parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] dplyr_0.8.3           cellrangerRkit_2.0.0 Rmisc_1.5
##  [4] plyr_1.8.4            lattice_0.20-38      bit64_0.9-7
##  [7] bit_1.1-14            RColorBrewer_1.1-2   monocle_2.10.1
## [10] DDRTree_0.1.5         irlba_2.3.3          VGAM_1.1-1
## [13] ggplot2_3.2.1         Biobase_2.42.0       BiocGenerics_0.28.0
## [16] Matrix_1.2-17
##
## loaded via a namespace (and not attached):
##  [1] viridis_0.5.1         viridisLite_0.3.0    assertthat_0.2.1
##  [4] yaml_2.2.0            slam_0.1-45          ggrepel_0.8.1
##  [7] pillar_1.4.2         glue_1.3.1           limma_3.38.3
## [10] densityClust_0.3     digest_0.6.21        colorspace_1.4-1
## [13] fastICA_1.2-2        htmltools_0.3.6      pkgconfig_2.0.3
## [16] pheatmap_1.0.12      HSMMSingleCell_1.2.0 qlcMatrix_0.9.7
## [19] purrr_0.3.2          scales_1.0.0         RANN_2.6.1
## [22] Rtsne_0.15           proxy_0.4-23         tibble_2.1.3
## [25] combinat_0.0-8       docopt_0.6.1         withr_2.1.2
## [28] sparsesvd_0.2        lazyeval_0.2.2       magrittr_1.5
## [31] crayon_1.3.4         evaluate_0.14        FNN_1.1.3
## [34] tools_3.5.0          data.table_1.12.2    matrixStats_0.55.0
## [37] stringr_1.4.0        Rhdf5lib_1.4.3       munsell_0.5.0
## [40] cluster_2.1.0        compiler_3.5.0       rlang_0.4.0
## [43] rhdf5_2.26.2         grid_3.5.0           rstudioapi_0.10
## [46] igraph_1.2.4.1       labeling_0.3         rmarkdown_1.15
## [49] gtable_0.3.0         codetools_0.2-16     reshape2_1.4.3
## [52] R6_2.4.0             gridExtra_2.3        knitr_1.25
## [55] stringi_1.4.3        Rcpp_1.0.2           tidyselect_0.2.5
## [58] xfun_0.9
```