PAPER TITLE TO BE DEFINED (in common.yaml)

10-DE genes across pseudotime

2021-12-09 14:07:55 +0100

Abstract

Lung interstitium macrophages (IMs) are non-alveolar resident tissue macrophages which contribute to the lung homeostasis. These cells were reported to be heterogeneous by our group and other teams, which contains two main distinct subpopulations: CD206+ IMs and CD206- IMs. However, the exact origin of IMs and the transcriptional programs that control IM differentiation remains unclear. In recent report, we analyzed the refilled IMs in the course of time after induced IM depletion with single-cell RNA sequencing (10X Genomics Chromium) and bulk RNA sequencing.

Contents

1 Description

2 Prepare data

```
suppressMessages(
{
library(Seurat)
library(ComplexHeatmap)
library(ggplot2)
library(dplyr)
library(RColorBrewer)
library(circlize)
library(monocle3)
})

mo <- readRDS(file = "../9-Monocle_analysis_and_pseudotime_estimation/Mono_to_IM.cds")
```

3 DE gene expression across IM-differentiation

DE genes across pseudotime of IM differentiation ## Across pseudotime of IM differentiation

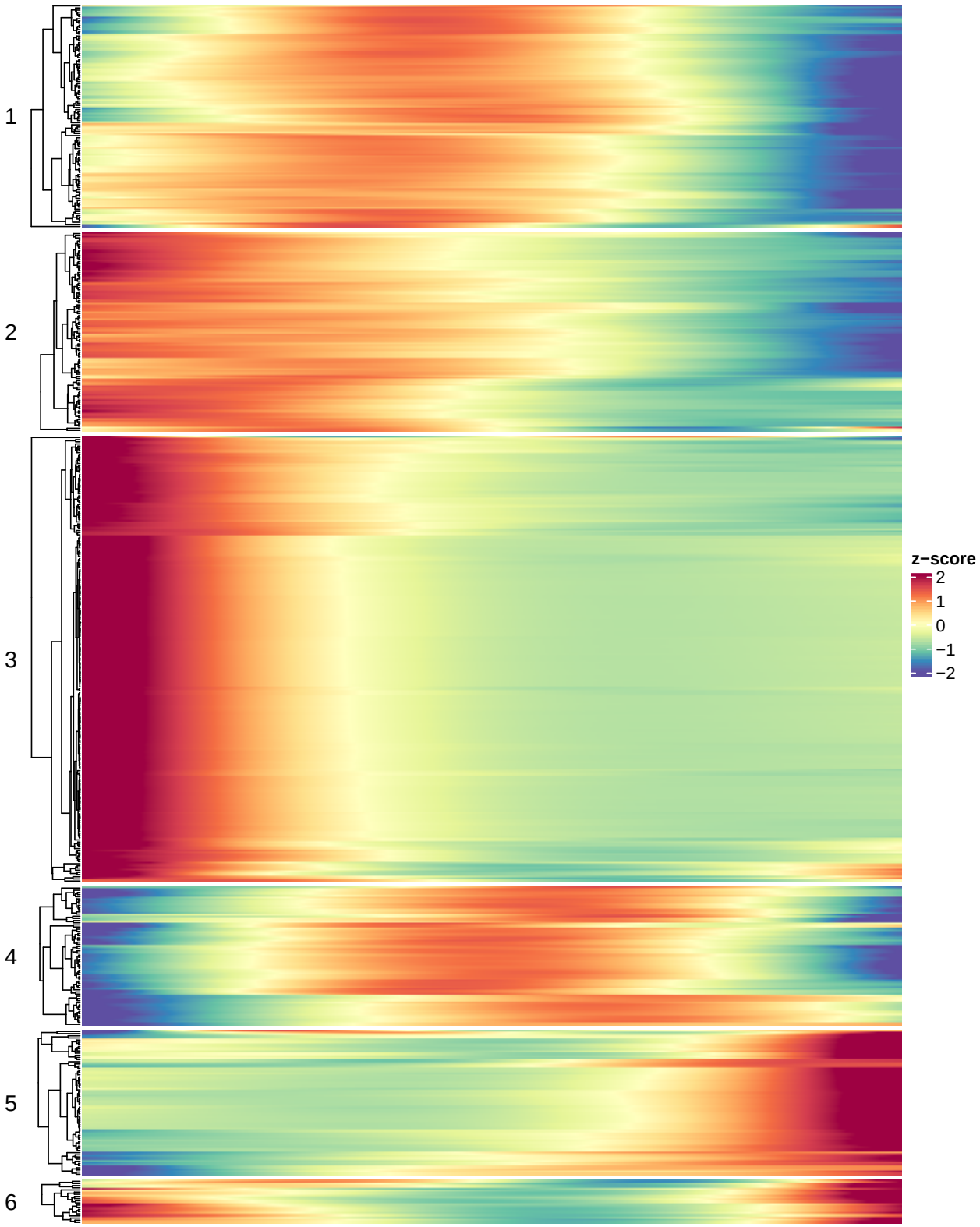
Prepare matrix with z-scores, smoothened and scaled data across pseudotime for heatmap.

```
pt.matrix <- exprs(mo)[match(genes,rownames(rowData(mo))),order(pseudotime(mo))]
cellnames <- colnames(pt.matrix)
#Can also use "normalized_counts" instead of "exprs" to use various normalization methods, for example:
#normalized_counts(cds, norm_method = "log")

pt.matrix <- t(apply(pt.matrix,1,function(x){smooth.spline(x,df=3)$y}))
pt.matrix <- t(apply(pt.matrix,1,function(x){(x-mean(x))/sd(x)}))
rownames(pt.matrix) <- genes
colnames(pt.matrix) <- cellnames
```

Show DE genes in unsupervised heatmap.

```
#K means with 6 groups
htkm <- Heatmap(
  pt.matrix,
  # use_raster = FALSE, # use FALSE to export to vector image.
  name                         = "z-score",
  col                          = colorRamp2(seq(from=-2,to=2,length=11),rev(brewer.pal(11, "Spectral"))),
  show_row_names               = FALSE,
  show_column_names            = FALSE,
  row_names_gp                 = gpar(fontsize = 3),
  row_km = 6,
  row_km_repeats = 31,
  row_dend_reorder = TRUE,
  row_title_rot                = 0,
  cluster_rows                 = TRUE,
  cluster_row_slices           = FALSE,
```

```
  cluster_columns               = FALSE,
)

htkm <- draw(htkm)
```



In this heatmap, the x axis is pseudotime, which represents differentiation state from monocytes (left) to IMs (right).

3.1 Annotate the cells associated to either differentiation of CD206+ IMs or CD206- IMs

```
library(magrittr)
# Get the closest vertice for every cell
y_to_cells <-  mo@principal_graph_aux$UMAP$pr_graph_cell_proj_closest_vertex%>%as.data.frame()

y_to_cells$cells <- rownames(y_to_cells)
y_to_cells$Y <- y_to_cells$V1


# Get the root vertices
# It is the same node as above
root <- mo@principal_graph_aux$UMAP$root_pr_nodes

principalgraph <- mo@principal_graph$UMAP

# Get the other endpoints
endpoints <- names(which(igraph::degree(principalgraph ) == 1))
endpoints <- endpoints[!endpoints %in% root]

# For each endpoint
cellWeights <- lapply(endpoints, function(endpoint) {
  # We find the path between the endpoint and the root
  path <- igraph::shortest_paths(principalgraph, root, endpoint)$vpath[[1]]
  path <- as.character(path)
  # We find the cells that map along that path
  df <- y_to_cells[y_to_cells$Y %in% path, ]
  df <- data.frame(weights = as.numeric(colnames(mo) %in% df$cells))
  colnames(df) <- endpoint
  return(df)
  }) %>% do.call(what = 'cbind', args = .) %>%
    as.matrix()
rownames(cellWeights) <- colnames(mo)
colnames(cellWeights) <- c("CD206_IM␣branch", "MHCII_IM␣branch")
pseudotime <- matrix(mo@principal_graph_aux$UMAP$pseudotime, ncol = ncol(cellWeights),
                     nrow = ncol(mo), byrow = FALSE)
rownames(pseudotime) <- colnames(mo)
```

4 TradeSeq analysis for the differentiation of monocytes to either of IM subsets

4.1 Construct sce object for TradeSeq

```
suppressMessages(library(tradeSeq))
```

```
# this step is VERY time consuming
sce <- fitGAM(counts = mo@assays@data$counts,
              pseudotime = pseudotime,
              cellWeights = cellWeights)
```

```

saveRDS(sce, file = "./sce.4339cells.newversion.Rds")
```

Between-lineage comparisons (CD206+ IM vs CD206- IM differentiation) ## Between-lineage comparisons (CD206+ IM vs CD206- IM differentiation)

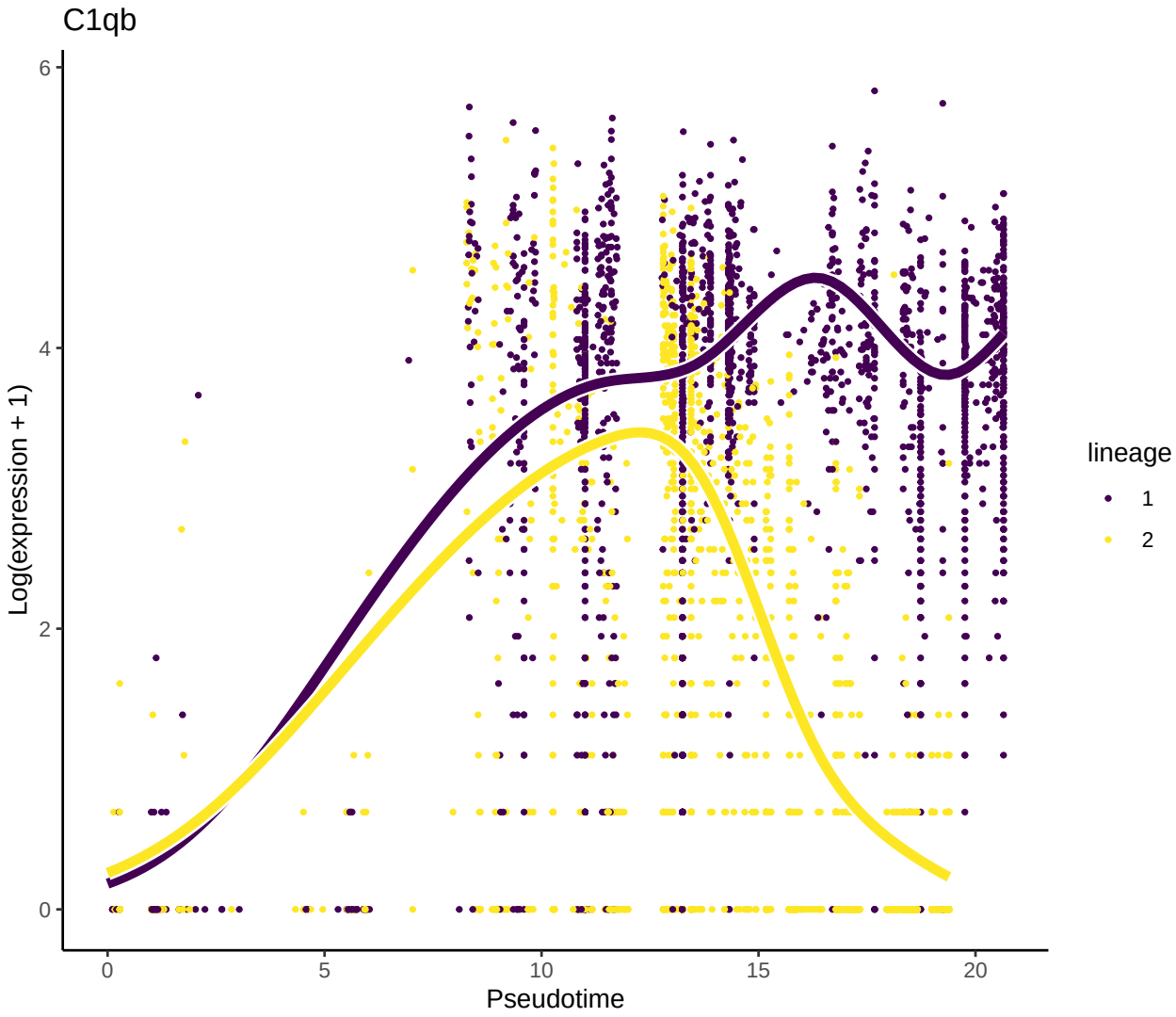
Association of gene expression with pseudotime (find significant DE genes along pseudotime).

```
assoRes <- associationTest(sce)
endRes <- diffEndTest(sce)
head(assoRes)
```

```
## # A tibble: 6 x 4
##   waldStat    df        pvalue meanLogFC
##      <dbl> <dbl>         <dbl>     <dbl>
## 1    210.      9  0               0.222
## 2     28.4     9  0.000815        0.113
## 3     NA      NA  NA              0.121
## 4     41.8     9  0.00000360      0.0958
## 5     36.4     9  0.0000330       0.163
## 6     45.1     9  0.000000880     0.160
```

Plot the most sig gene:

```
library(ggplot2)
o <- order(endRes$waldStat, decreasing = TRUE)
sigGene <- names(sce)[o[1]]
plotSmoothers(sce, counts = counts(sce), gene = sigGene
              #, curvesCol = c("#33A02C", "#B2DF8A")
              ) + ggtitle(sigGene)
```



What's the top genes?

```
names(sce)[o[1:20]]
```

```
##  [1] "C1qb"     "Ctsb"     "C1qa"     "Selenop"  "Csf1r"    "Timp2"    "Pf4"
##  [8] "C1qc"     "Serinc3"  "Cd209a"   "Lsp1"     "Lgmn"     "Apoe"     "Blvrb"
    "
## [15] "Olfm1"    "Tnip3"    "Rpl13"    "Ninj1"    "Rpl28"    "H2-DMb1"
```

4.2 Clustering using RSEC, clusterExperiment

tradeSeq provides the functionality to cluster genes according to their expression pattern along the lineages with the clusterExpressionPatterns function. A number of equally spaced points for every lineage are selected to perform the clustering, and the number of points can be selected with the nPoints argument. (from `vignette("tradeSeq")`)

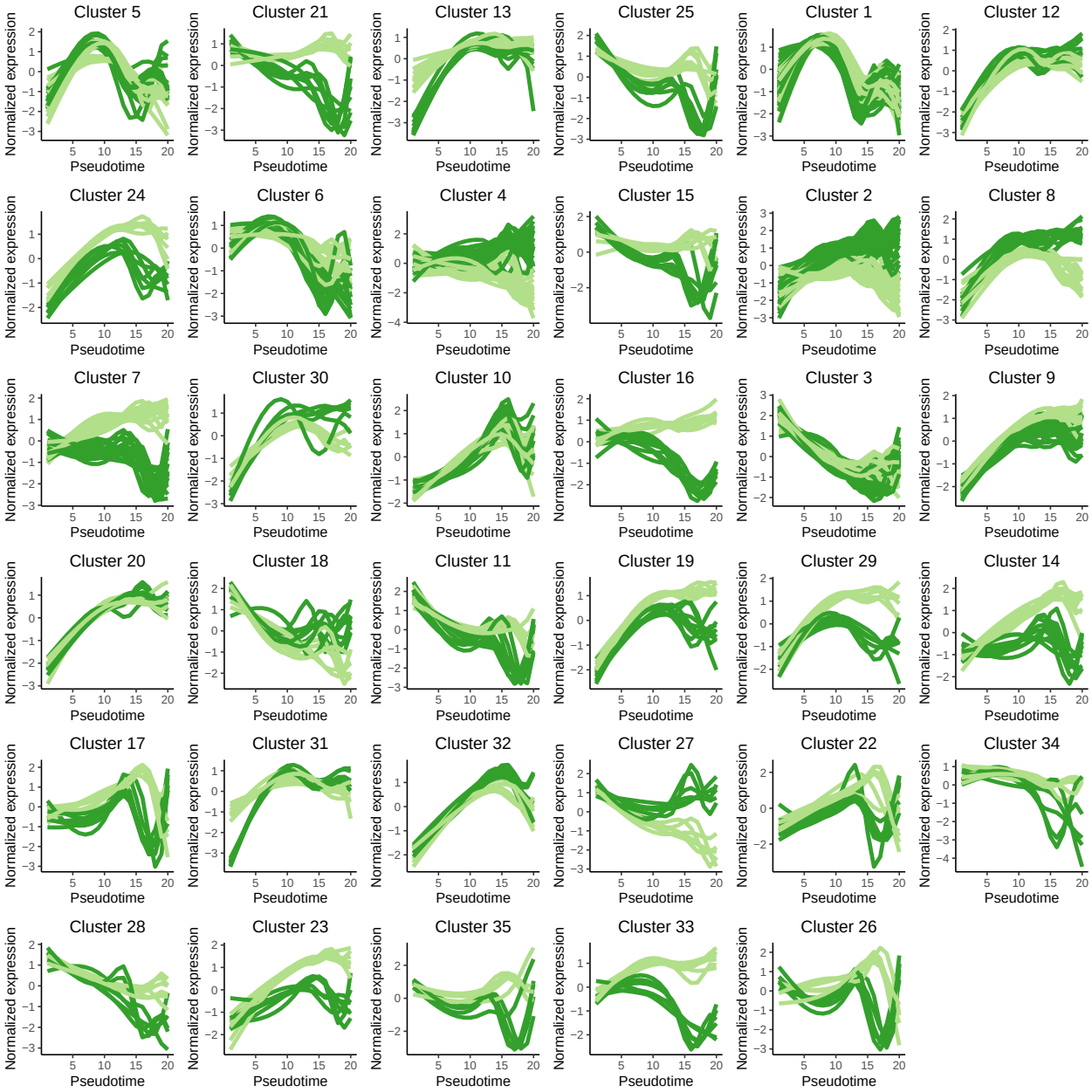
```
library(clusterExperiment)
nPointsClus <- 20 # The number of points to use for clustering the
    expression patterns..
clusPat <- clusterExpressionPatterns(sce,
```

```
                                    nPoints = nPointsClus,
                                    genes = genes,
                                    random.seed = 43,
                                    beta = 0.2
                                    )
```

```
## 36 parameter combinations, 36 use sequential method, 36 use subsampling method
## Running Clustering on Parameter Combinations...
## done.
```

```
clusterLabels <- primaryCluster(clusPat$rsec)
```

```
cUniq <- unique(clusterLabels) #
cUniq <- cUniq[!cUniq == -1] # remove unclustered genes

# cUniq <- cUniq[cUniq == -1]
#Any samples not found as part of a homogenous set of clusters at that point will be classified as unclustered (given a value of -1)

# beta: value between 0 and 1 to decide how stable clustership membership has to be before 'finding' and removing the cluster.
if (exists("p.total")) { rm(p.total)}

for (xx in cUniq) {
  cId <- which(clusterLabels == xx)
  p <- ggplot(data = data.frame(x = 1:nPointsClus,
                                y = rep(range(clusPat$yhatScaled[cId, ]),
                                        nPointsClus / 2)),
              aes(x = x, y = y)) +
    geom_point(alpha = 0) +
    labs(title = paste0("Cluster␣", xx),  x = "Pseudotime", y = "Normalized␣expression") +
    theme_classic() +
    theme(plot.title = element_text(hjust = 0.5))
  for (ii in 1:length(cId)) {
    geneId <- rownames(clusPat$yhatScaled)[cId[ii]]
    p <- p +
      geom_line(data = data.frame(x = rep(1:nPointsClus, 2),
                                  y = clusPat$yhatScaled[geneId, ],
                                  lineage = rep(0:1, each = nPointsClus)),
                aes(col = as.character(lineage), group = lineage), lwd = 1.5)
  }
  p <- p + guides(color = "none") +
    scale_color_manual(values = c("#33A02C", "#B2DF8A"),
                       breaks = c("0", "1"))
  if (exists("p.total")) { p.total <- p.total + p} else {p.total <- p}
}
  print(p.total)
```



5 Show gene expression pattern calculated by TradeSeq in heatmap

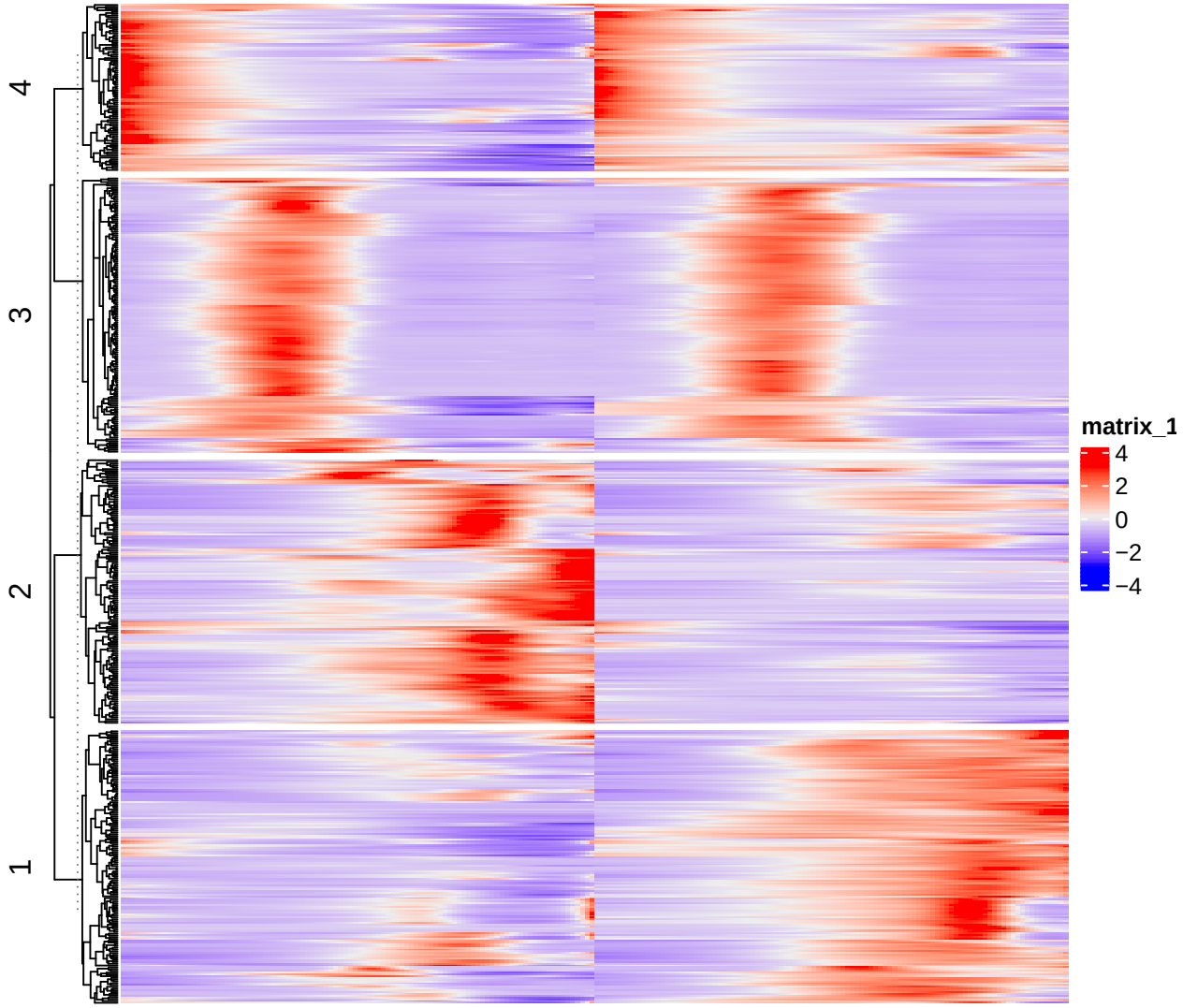
5.1 Data preparation

Here we use the DE genes calculated in DE genes across pseudotime of IM differentiation.

```
yhatSmooth <- predictSmooth(sce, gene = genes, nPoints = 100, tidy = FALSE
    )
yhatSmoothScaled <- t(scale(t(yhatSmooth)))
```

5.2 Draw heatmap

```
heatSmooth <- Heatmap(yhatSmoothScaled, cluster_columns = FALSE, show_row_
   names = FALSE, show_column_names = FALSE, row_km = 4)
heatSmooth <- draw(heatSmooth)
```



Two IM differentiation show similar patterns but some genes (especially cluster 2 and 1) are different in CD206+ and CD206-.

5.3 Annotate DE genes as CD206+/CD206- IM differentiation specific or common genes

According to the heatmap above, some of DE genes should remain unchanged (common) and half of them are specific to one of two IM differentiation.

We use wald statistic calculated in diffEndTest to annotate the "common" genes and "specific" genes. (in Between-lineage comparisons (CD206+ IM vs CD206- IM differentiation))

```
endRes.DE <- endRes[rownames(yhatSmooth), ]
summary(endRes.DE$waldStat)
```

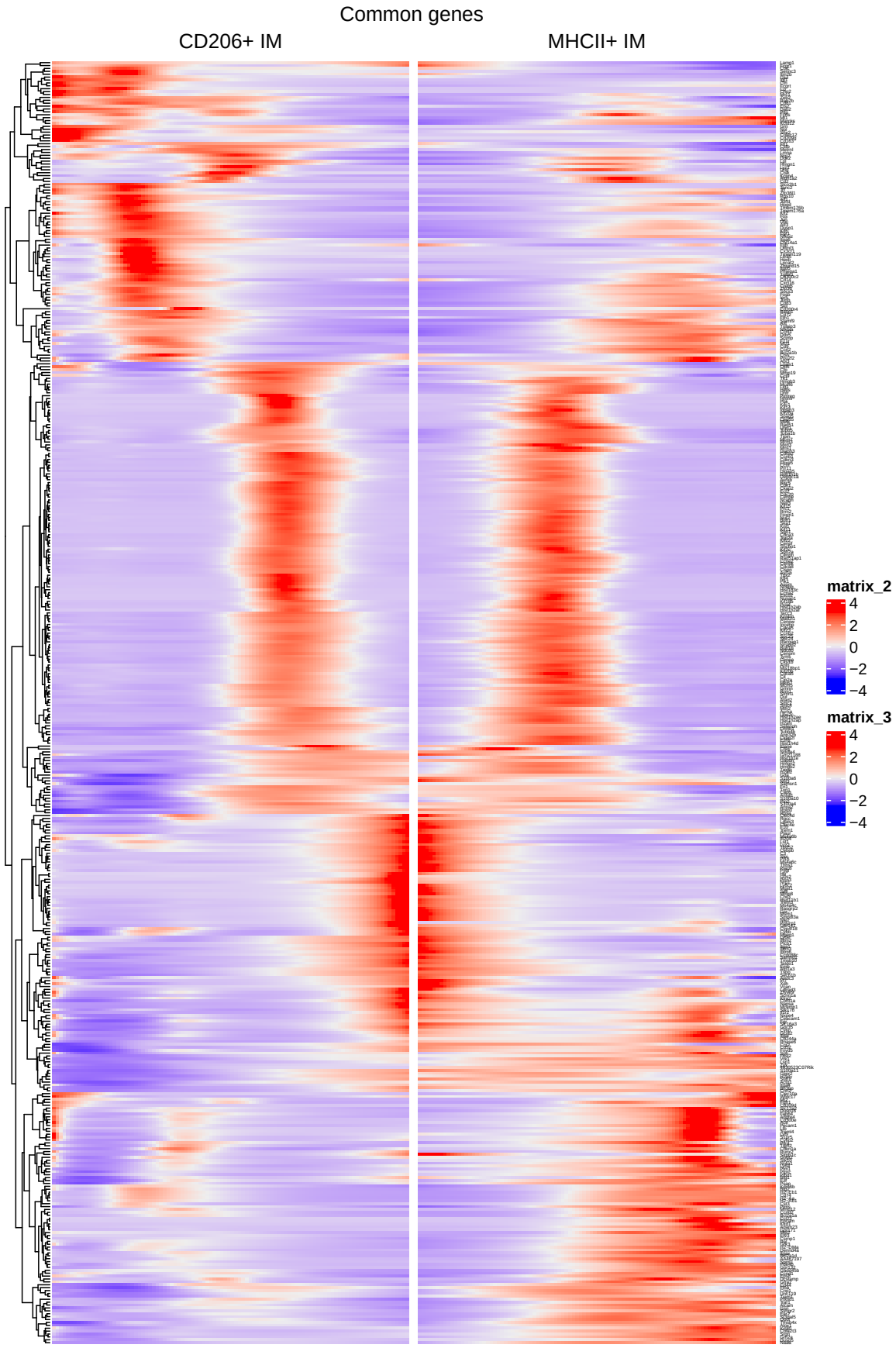
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
##     0.0027     3.8599    41.4571   123.7353   168.7434 1263.9587
```

Let's use waldStat > 40 and logFC > 2 as cut threshold.

```
genes.changed <- rownames(filter(endRes.DE, waldStat > 70 & (logFC1_2 > 2
    | logFC1_2 < -2) ))
genes.noChange <- setdiff(rownames(endRes.DE) , genes.changed)
```

Make heatmap with unchanged/common genes.

```
heatSmooth_cd206.unchanged <- Heatmap(yhatSmoothScaled[genes.noChange,
    100:1], cluster_columns = FALSE,  show_row_names = FALSE, show_column_
    names = FALSE, column_title = "CD206+␣IM")

heatSmooth_mhcii.unchanged <- Heatmap(yhatSmoothScaled[genes.noChange,
    101:200], cluster_columns = FALSE,  show_row_names = TRUE, row_names_gp
     = gpar(fontsize = 3), show_column_names = FALSE, column_title = "MHCII
    +␣IM")

heatSmooth_combined.unchanged <- draw ( heatSmooth_cd206.unchanged +
    heatSmooth_mhcii.unchanged, column_title = "Common␣genes", auto_adjust
    = FALSE)
```


Common genes
CD206+ IM
MHCII+ IM
matrix_2
4
2
0
−2
−4
matrix_3
4
2
0
−2
−4

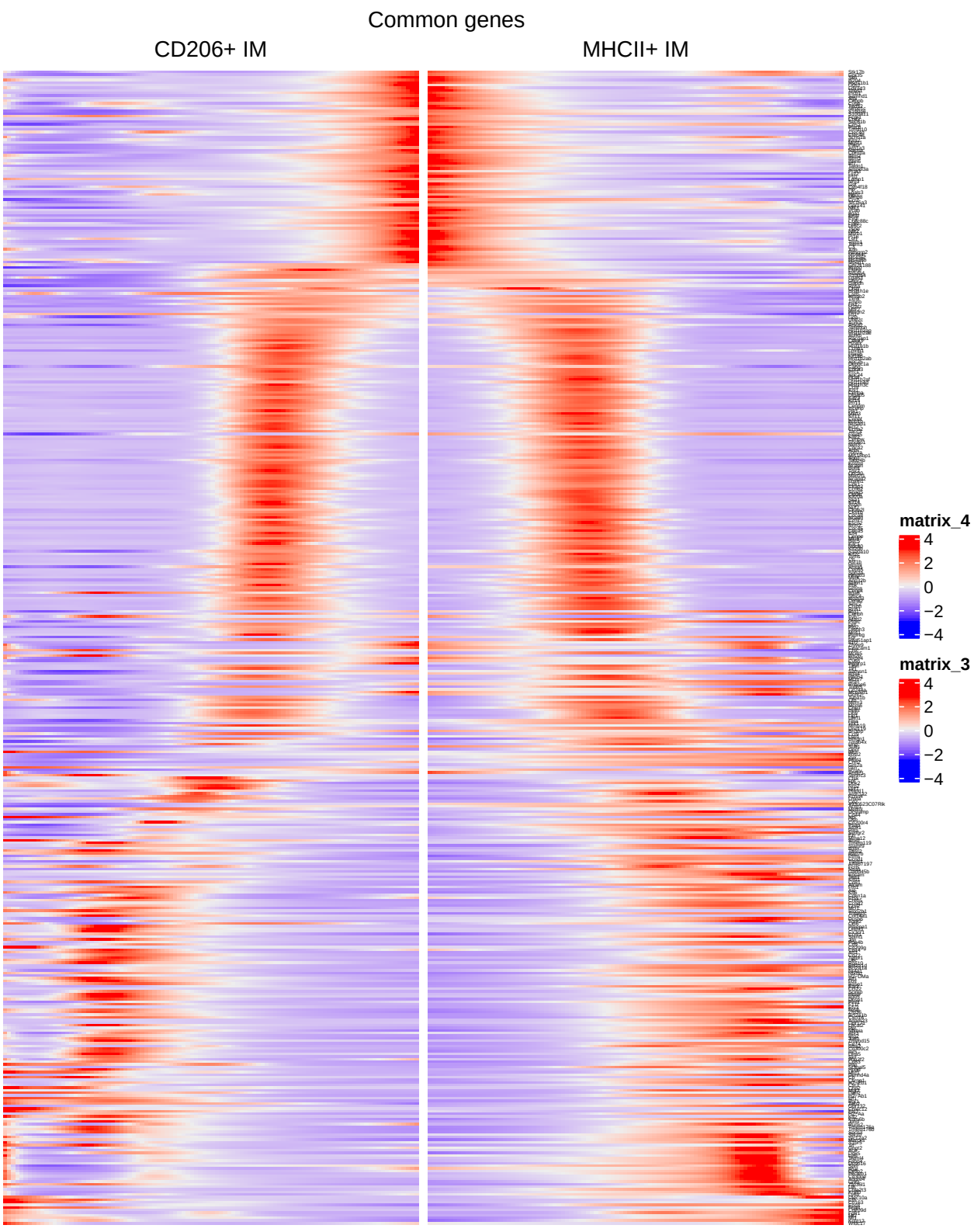
5.4 Daw heatmap with expression patterns of unchanged/common genes in the order of pseudotime

Let's find the expression peak of each gene:

```
orderbyExpressionPeak <- function(x, # matrix
                                  decreasing = FALSE,
                                  output.position = FALSE # if true, give
                                     relative position 0 - 1, or output
                                     order.
                                  ) {
  indx.peak <- apply(x, 1 , which.max)

  if(output.position) {
    po <- indx.peak/nrow(x)
    if (! length(rownames(x)) == 0) {names(po) <- rownames(x)}
    return(po)
  } else {
    o <- order(indx.peak)
    if (! length(rownames(x)) == 0) {names(o) <- rownames(x)[o]}
    return (o)}
}
```

Make average peak pseudotime peak for each gene:

```
po.cd206 <- orderbyExpressionPeak(yhatSmoothScaled[genes.noChange, 1:100],
    output.position = TRUE)
po.mhcii <- orderbyExpressionPeak(yhatSmoothScaled[genes.noChange,
   101:200], output.position = TRUE)
order.mean <- order ( ( po.cd206 + po.mhcii ) /2)

heatSmooth_cd206.unchanged.ordered <- Heatmap(yhatSmoothScaled[genes.
   noChange, 100:1], cluster_columns = FALSE,  show_row_names = FALSE,
   show_column_names = FALSE, row_order = order.mean,  column_title = "
   CD206+ IM")

heatSmooth_combined.unchanged.ordered <- draw ( heatSmooth_cd206.unchanged
   .ordered + heatSmooth_mhcii.unchanged, column_title = "Common genes",
   auto_adjust = FALSE)
```

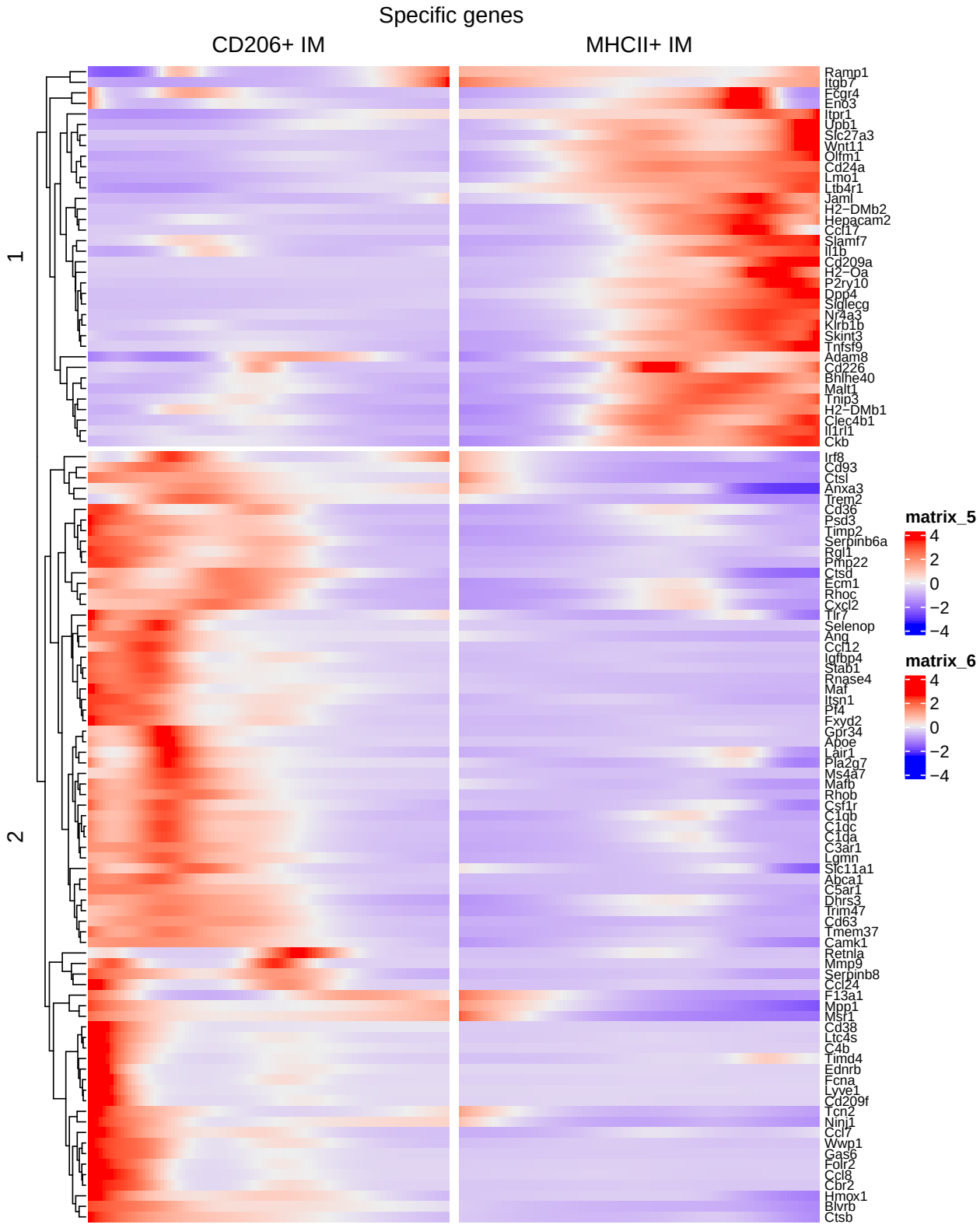



```
pdf(file = "../Figures/Heatmap_common_genes_IMs_diff_across_pseudotime.pdf
   ", width = 8, height = 10)
heatSmooth_combined.unchanged.ordered
dev.off()
```

```
## pdf
##   2
```

5.5 Make with changed/specific genes

```
heatSmooth_cd206.changed <- Heatmap(yhatSmoothScaled[genes.changed,
   100:1], cluster_columns = FALSE,  show_row_names = FALSE, cluster_rows
   = hclust(dist(yhatSmoothScaled[genes.changed, ])), show_column_names =
   FALSE, column_title = "CD206+ IM")

heatSmooth_mhcii.changed <- Heatmap(yhatSmoothScaled[genes.changed,
   101:200], cluster_columns = FALSE,  show_row_names = TRUE,row_names_gp
   = gpar(fontsize = 8), show_column_names = FALSE, column_title = "MHCII+
   IM")

heatSmooth_combined.changed <- draw ( heatSmooth_cd206.changed +
   heatSmooth_mhcii.changed, column_title = "Specific genes", split = 2)
```



6 Functionality analysis of DE genes across pseudotime (common genes)

```
common.genes <- heatSmooth_cd206.unchanged.ordered@row_names_param$labels[
   heatSmooth_cd206.unchanged.ordered@row_order]
```

6.1 Manually classify genes by expression timing peak

Class-1 genes are the genes expressed in monocytes but turned off in the very early phase:

```
genes.class1 <- common.genes[1:which(common.genes == "Gm21188")]
length(genes.class1)
```

```
## [1] 75
```

Class-2 genes are the genes up-regulated in early phase of differentiation and turned off during transit phase.

```
genes.class2 <- common.genes[(which(common.genes == "Gm21188")+1) : which(
   common.genes == "Diaph3")]
length(genes.class2)
```

```
## [1] 140
```

Class-3 genes are the late upregulated genes during IM differentiation.

```
genes.class3 <- common.genes[(which(common.genes == "Diaph3")+1) : length(
   common.genes)]
length(genes.class3)
```

```
## [1] 228
```

Save gene lists:

```
write.csv(genes.class1, file = "./common_genes_class1.csv", quote = FALSE)
write.csv(genes.class2, file = "./common_genes_class2.csv", quote = FALSE)
write.csv(genes.class3, file = "./common_genes_class3.csv", quote = FALSE)
```

6.2 GO/KEGG enrichment analysis with 3 classes of common genes

```
suppressMessages(library(clusterProfiler))
source("../R/entrez2symbol.R")
source("../R/replaceEntrezID.R")
```

6.2.1 KEGG enrichment for common genes class 1

```
symb <- genes.class1
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
   , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
result.enrichKEGG <- enrichKEGG(de_entrez, organism = "mmu", keyType = "
   ncbi-geneid")
result.enrichKEGG <- replaceEntrezID(result.enrichKEGG, organism = "mmu")
write.csv(result.enrichKEGG@result, file = "./Results_enrichment/
   enrichKEGG_common_genes_class1.csv")
result.enrichKEGG@result
```

```
## # A tibble: 149 x 9                                                    1
##    ID       Description  GeneRatio BgRatio  pvalue p.adjust  qvalue     2
   geneID Count
##    <chr>    <chr>        <chr>     <chr>     <dbl>    <dbl>   <dbl> <    3
   chr>  <int>
##  1 mmu04145 Phagosome    7/36      182/89~ 6.53e-6 0.000973 9.00e-4     4
   Thbs1~     7
##  2 mmu05152 Tuberculosis 6/36      180/89~ 7.21e-5 0.00537  4.97e-3     5
   Cebpb~     6
##  3 mmu05140 Leishmanias~ 3/36      70/8943 2.73e-3 0.119    1.10e-1     6
   Cybb/~     3
##  4 mmu04918 Thyroid hor~ 3/36      74/8943 3.19e-3 0.119    1.10e-1     7
   Plcb1~     3
##  5 mmu04970 Salivary se~ 3/36      86/8943 4.88e-3 0.145    1.34e-1     8
   Plcb1~     3
##  6 mmu04610 Complement ~ 3/36      93/8943 6.07e-3 0.151    1.39e-1     9
   Plaur~     3
##  7 mmu04613 Neutrophil ~ 4/36      207/89~ 9.20e-3 0.167    1.55e-1     10
   Plcb1~     4
##  8 mmu04621 NOD-like re~ 4/36      211/89~ 9.82e-3 0.167    1.55e-1     11
   Ifi20~     4
##  9 mmu04960 Aldosterone~ 2/36      38/8943 1.01e-2 0.167    1.55e-1     12
   Scnn1~     2
## 10 mmu04973 Carbohydrat~ 2/36      48/8943 1.58e-2 0.236    2.18e-1     13
   Plcb1~     2
## # ... with 139 more rows                                               14
```

6.2.2 GO enrichment for common genes class 1

```
result.enrichGO <- enrichGO(de_entrez, OrgDb = "org.Mm.eg.db", ont = "BP")  1
result.enrichGO <- replaceEntrezID(result.enrichGO, organism = "mmu")       2
write.csv(result.enrichGO@result, file = "./Results_enrichment/enrichGO_    3
   common_genes_class1.csv")
result.enrichGO@result                                                      4
```

```
## # A tibble: 1,740 x 9                                                  1
##    ID     Description  GeneRatio BgRatio  pvalue p.adjust  qvalue       2
   geneID   Count
##    <chr>  <chr>        <chr>     <chr>     <dbl>    <dbl>   <dbl> <chr>  3
       <int>
##  1 GO:00~ myeloid leu~ 10/72     219/23~ 1.40e-9  2.16e-6 1.65e-6 Gpr35  4
   /S~    10
##  2 GO:00~ cellular ex~ 7/72      72/233~ 2.48e-9  2.16e-6 1.65e-6 Sell/  5
   Pl~     7
##  3 GO:00~ leukocyte m~ 11/72     360/23~ 1.32e-8  7.68e-6 5.86e-6 Gpr35  6
   /S~    11
##  4 GO:00~ leukocyte c~ 9/72      219/23~ 2.45e-8  1.07e-5 8.14e-6 Gpr35  7
   /S~     9
##  5 GO:00~ positive re~ 11/72     418/23~ 6.08e-8  2.12e-5 1.61e-5 Sell/  8
   If~    11
##  6 GO:00~ type I inte~ 5/72      40/233~ 1.47e-7  3.64e-5 2.78e-5        9
   Samhd1/~     5
```

```
##  7 GO:00~ cellular re~ 5/72      40/233~ 1.47e-7  3.64e-5 2.78e-5
   Samhd1/~     5
##  8 GO:00~ defense res~ 11/72     464/23~ 1.74e-7  3.64e-5 2.78e-5 Slpi/
   Ce~    11
##  9 GO:00~ positive re~ 9/72      278/23~ 1.88e-7  3.64e-5 2.78e-5
   Ifi204/~     9
## 10 GO:00~ response to~ 5/72      45/233~ 2.70e-7  4.70e-5 3.59e-5
   Samhd1/~     5
## # ... with 1,730 more rows
```

6.2.3 KEGG enrichment for common genes class 2

```
symb <- genes.class2
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
   , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
result.enrichKEGG <- enrichKEGG(de_entrez, organism = "mmu", keyType = "
   ncbi-geneid")
result.enrichKEGG <- replaceEntrezID(result.enrichKEGG, organism = "mmu")
write.csv(result.enrichKEGG@result, file = "./Results_enrichment/
   enrichKEGG_common_genes_class2.csv")
result.enrichKEGG@result
```

```
## # A tibble: 88 x 9
##    ID       Description GeneRatio BgRatio   pvalue p.adjust  qvalue
   geneID Count
##    <chr>    <chr>       <chr>     <chr>      <dbl>    <dbl>   <dbl> <
   chr>  <int>
##  1 mmu04114 Oocyte mei~ 11/50     121/89~ 4.20e-11  2.64e-9 2.28e-9
   Aurka~    11
##  2 mmu04110 Cell cycle  11/50     125/89~ 6.01e-11  2.64e-9 2.28e-9
   Ccnb1~    11
##  3 mmu04914 Progestero~ 9/50      92/8943 1.54e- 9  4.51e-8 3.88e-8
   Aurka~     9
##  4 mmu04115 p53 signal~ 5/50      72/8943 4.70e- 5  1.03e-3 8.90e-4
   Ccnb1~     5
##  5 mmu00240 Pyrimidine~ 4/50      58/8943 2.94e- 4  5.17e-3 4.45e-3
   Rrm2/~     4
##  6 mmu04218 Cellular s~ 6/50      184/89~ 5.24e- 4  7.68e-3 6.62e-3
   Ccnb1~     6
##  7 mmu05222 Small cell~ 4/50      93/8943 1.75e- 3  2.20e-2 1.90e-2 Fn1
   /C~     4
##  8 mmu05166 Human T-ce~ 6/50      250/89~ 2.55e- 3  2.64e-2 2.28e-2
   Bub1b~     6
##  9 mmu05132 Salmonella~ 6/50      253/89~ 2.70e- 3  2.64e-2 2.28e-2
   Gapdh~     6
## 10 mmu04512 ECM-recept~ 3/50      88/8943 1.29e- 2  1.14e-1 9.80e-2 Fn1
   /H~     3
## # ... with 78 more rows
```

6.2.4 GO enrichment for common genes class 2

```
result.enrichGO <- enrichGO(de_entrez, OrgDb = "org.Mm.eg.db", ont = "BP")
```

```
result.enrichGO <- replaceEntrezID(result.enrichGO, organism = "mmu")
write.csv(result.enrichGO@result, file = "./Results_enrichment/enrichGO_
    common_genes_class2.csv")
result.enrichGO@result
```

```
## # A tibble: 1,780 x 9
##    ID     Description GeneRatio BgRatio   pvalue p.adjust   qvalue
    geneID  Count
##    <chr>  <chr>       <chr>     <chr>      <dbl>    <dbl>    <dbl> <chr
    >   <int>
##  1 GO:00~ chromosome~ 50/127    324/23~ 7.39e-60 1.32e-56 1.08e-56
    Ube2c/~    50
##  2 GO:00~ nuclear ch~ 43/127    262/23~ 2.75e-52 1.75e-49 1.44e-49
    Ube2c/~    43
##  3 GO:00~ sister chr~ 39/127    181/23~ 2.95e-52 1.75e-49 1.44e-49
    Ube2c/~    39
##  4 GO:01~ mitotic nu~ 43/127    268/23~ 7.78e-52 2.91e-49 2.39e-49
    Ube2c/~    43
##  5 GO:00~ mitotic si~ 37/127    151/23~ 8.18e-52 2.91e-49 2.39e-49
    Ube2c/~    37
##  6 GO:00~ nuclear di~ 46/127    418/23~ 9.43e-48 2.80e-45 2.30e-45
    Ube2c/~    46
##  7 GO:00~ organelle ~ 47/127    472/23~ 9.12e-47 2.32e-44 1.91e-44
    Mtfr2/~    47
##  8 GO:00~ spindle or~ 27/127    179/23~ 1.59e-31 3.54e-29 2.91e-29
    Aurka/~    27
##  9 GO:19~ microtubul~ 24/127    142/23~ 2.56e-29 5.06e-27 4.16e-27
    Aurka/~    24
## 10 GO:00~ regulation~ 21/127    103/23~ 1.42e-27 2.53e-25 2.08e-25
    Ube2c/~    21
## # ... with 1,770 more rows
```

6.2.5 KEGG enrichment for common genes class 3

```
symb <- genes.class3
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
    , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
result.enrichKEGG <- enrichKEGG(de_entrez, organism = "mmu", keyType = "
    ncbi-geneid")
result.enrichKEGG <- replaceEntrezID(result.enrichKEGG, organism = "mmu")
write.csv(result.enrichKEGG@result, file = "./Results_enrichment/
    enrichKEGG_common_genes_class3.csv")
result.enrichKEGG@result
```

```
## # A tibble: 228 x 9
##    ID       Description  GeneRatio BgRatio  pvalue p.adjust  qvalue
    geneID Count
##    <chr>    <chr>        <chr>     <chr>     <dbl>    <dbl>   <dbl> <
    chr>  <int>
##  1 mmu04210 Apoptosis    15/141    136/89~ 2.97e-9  6.77e-7 4.69e-7
    Ctsc/~    15
##  2 mmu04145 Phagosome    15/141    182/89~ 1.56e-7  1.78e-5 1.23e-5
    Tubb5~    15
```

```
##  3 mmu05166 Human T-cel~ 17/141    250/89~ 3.62e-7  2.27e-5 1.57e-5
   Il1r2~    17
##  4 mmu05202 Transcripti~ 16/141    223/89~ 3.97e-7  2.27e-5 1.57e-5
   Il1r2~    16
##  5 mmu04640 Hematopoiet~ 10/141    94/8943 2.02e-6  9.20e-5 6.37e-5
   Il1r2~    10
##  6 mmu04380 Osteoclast ~ 11/141    128/89~ 5.10e-6  1.94e-4 1.34e-4
   Fosl2~    11
##  7 mmu05323 Rheumatoid ~ 9/141     87/8943 8.42e-6  2.74e-4 1.90e-4
   Ctsk/~     9
##  8 mmu05140 Leishmanias~ 8/141     70/8943 1.30e-5  3.69e-4 2.56e-4
   Itga4~     8
##  9 mmu05152 Tuberculosis 12/141    180/89~ 2.55e-5  6.46e-4 4.48e-4
   Lsp1/~    12
## 10 mmu04064 NF-kappa B ~ 9/141     105/89~ 3.88e-5  8.85e-4 6.13e-4
   Gadd4~     9
## # ... with 218 more rows
```

6.2.6 GO enrichment for common genes class 3

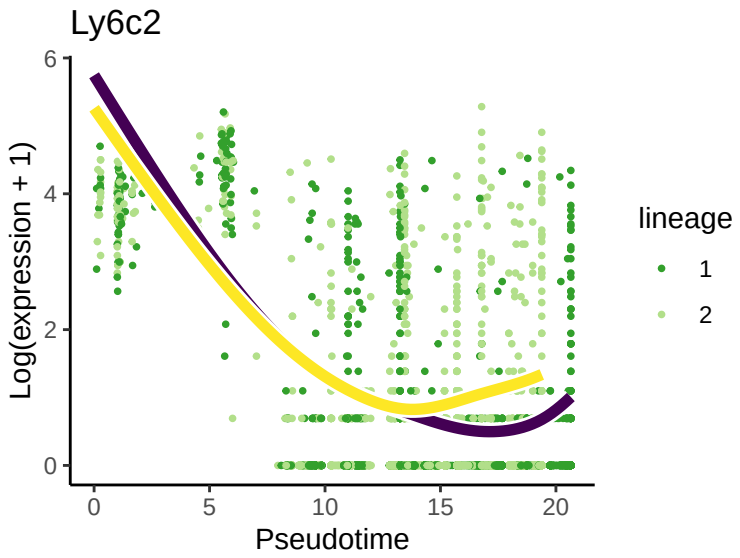
```
result.enrichGO <- enrichGO(de_entrez, OrgDb = "org.Mm.eg.db", ont = "BP")
result.enrichGO <- replaceEntrezID(result.enrichGO, organism = "mmu")
write.csv(result.enrichGO@result, file = "./Results_enrichment/enrichGO_
   common_genes_class3.csv")
result.enrichGO@result
```

```
## # A tibble: 3,419 x 9
##    ID     Description  GeneRatio BgRatio   pvalue p.adjust  qvalue
   geneID  Count
##    <chr>  <chr>        <chr>     <chr>      <dbl>    <dbl>   <dbl> <chr
   >   <int>
##  1 GO:00~ regulation ~ 23/224    372/23~ 1.72e-12  5.89e-9 4.30e-9
   Il1r2/~    23
##  2 GO:00~ negative re~ 22/224    462/23~ 8.04e-10  7.67e-7 5.61e-7 Fgr/
   Ce~    22
##  3 GO:00~ positive re~ 17/224    265/23~ 8.72e-10  7.67e-7 5.61e-7
   Ceacam~    17
##  4 GO:19~ regulation ~ 21/224    424/23~ 9.85e-10  7.67e-7 5.61e-7
   Ceacam~    21
##  5 GO:00~ leukocyte c~ 19/224    345/23~ 1.12e- 9  7.67e-7 5.61e-7
   Ceacam~    19
##  6 GO:00~ leukocyte m~ 19/224    360/23~ 2.27e- 9  1.29e-6 9.44e-7
   Itga4/~    19
##  7 GO:00~ regulation ~ 19/224    372/23~ 3.88e- 9  1.75e-6 1.28e-6
   Ceacam~    19
##  8 GO:19~ positive re~ 15/224    221/23~ 4.10e- 9  1.75e-6 1.28e-6
   Ceacam~    15
##  9 GO:00~ lymphocyte ~ 11/224    103/23~ 4.60e- 9  1.75e-6 1.28e-6
   Itga4/~    11
## 10 GO:00~ antigen pro~ 6/224     16/233~ 5.42e- 9  1.85e-6 1.35e-6 Ctss
   /H~     6
## # ... with 3,409 more rows
```

7 Show gene expression pattern with TradeSeq results

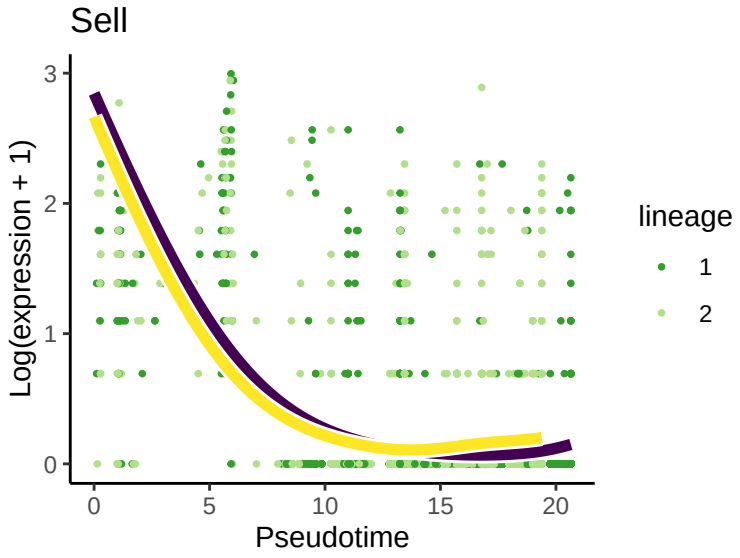
7.1 Class 1 common genes

```
require(ggplot2)
sigGene <- "Ly6c2"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```

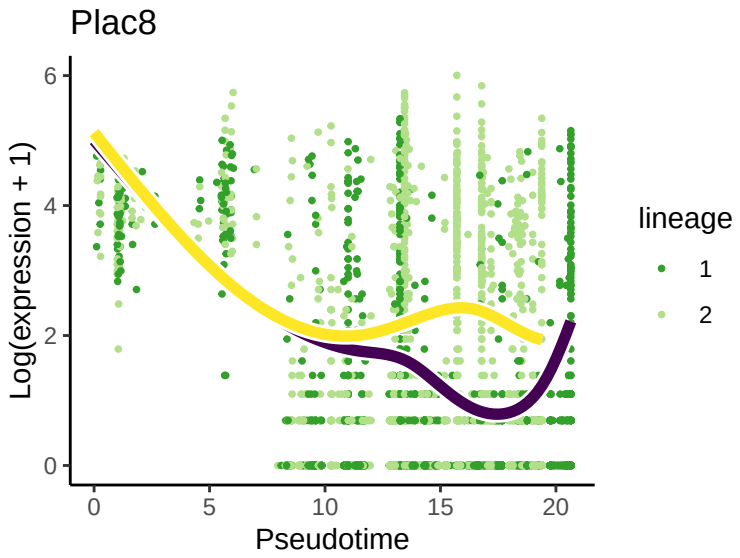



```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
       width = 4, height = 3)
```

```
sigGene <- "Sell"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```



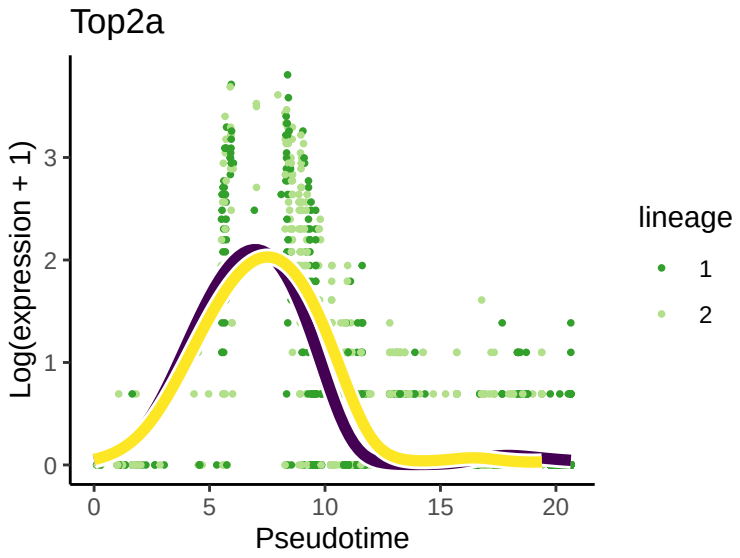
```
sigGene <- "Plac8"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
       width = 4, height = 3)
```

7.2 Class 2 common genes

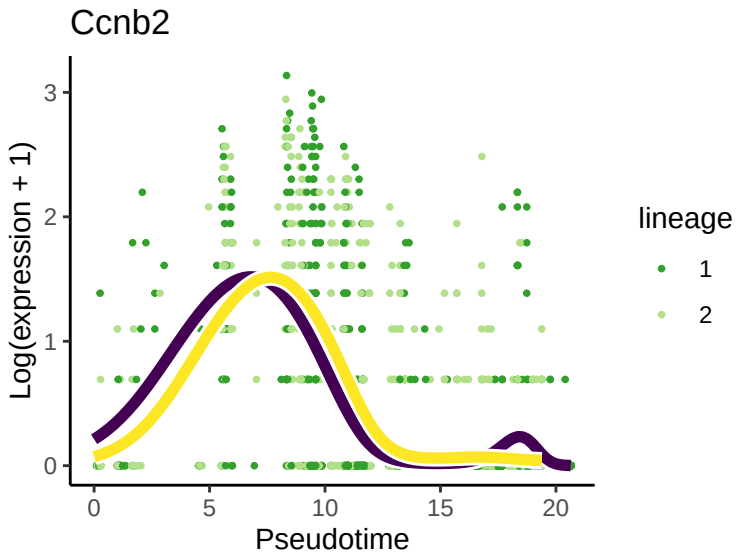
```
sigGene <- "Top2a"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
```

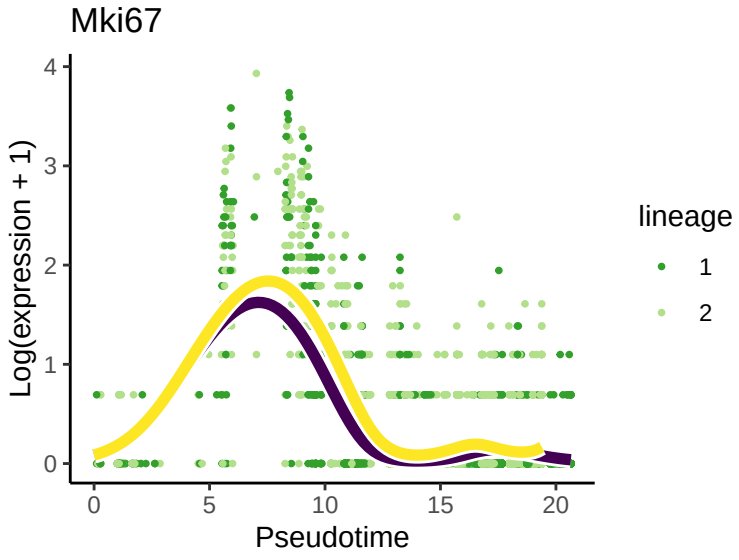
```
       width = 4, height = 3)
```

```
sigGene <- "Ccnb2"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
       width = 4, height = 3)
```

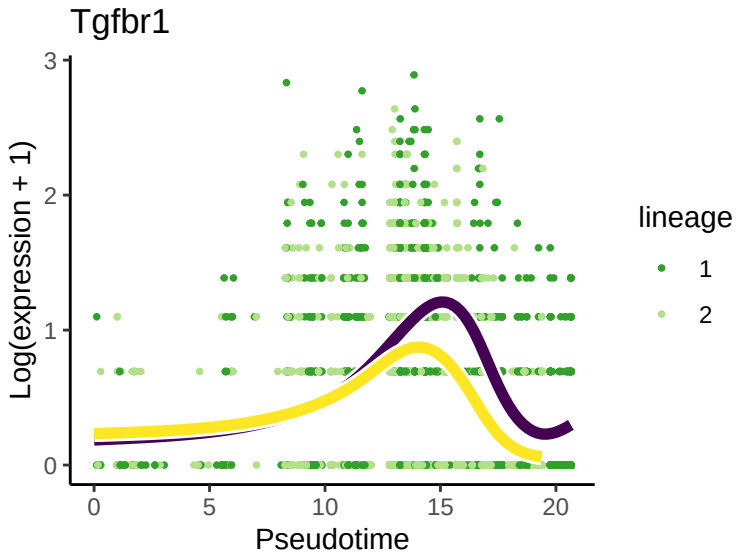
```
sigGene <- "Mki67"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
       width = 4, height = 3)
```

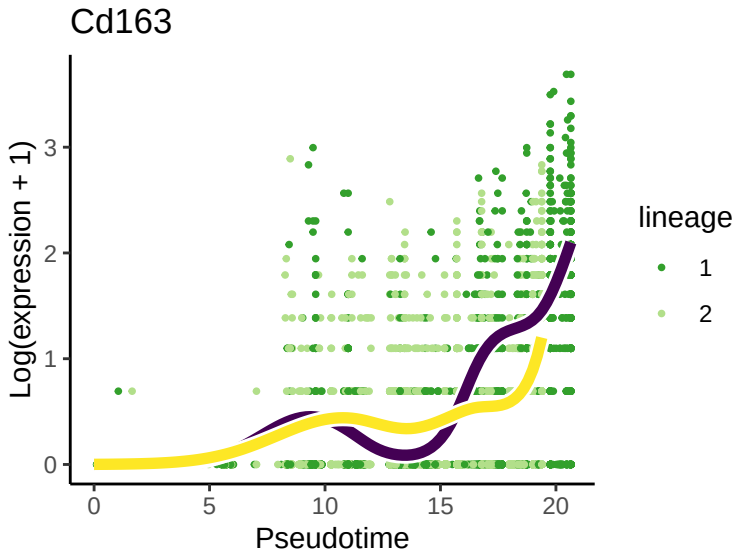
7.3 Class 3 common genes

```
sigGene <- "Tgfbr1"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```



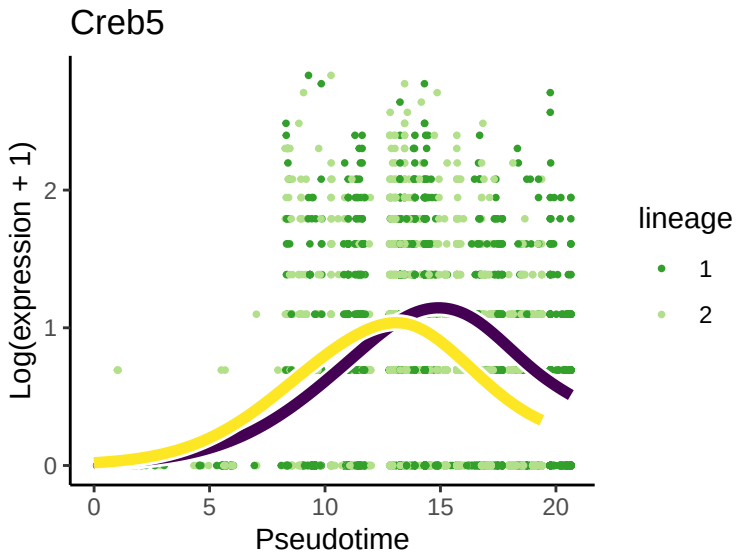

```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
        width = 4, height = 3)
```

```
sigGene <- "Cd163"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
        width = 4, height = 3)
```

```
sigGene <- "Creb5"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene
    ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```




```
ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_", sigGene,
    ".pdf"),
       width = 4, height = 3)
```

8 Session information

R sesssion:

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] stats4    parallel  grid      stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
```

```
## other attached packages:
##  [1] org.Mm.eg.db_3.12.0         AnnotationDbi_1.52.0
##  [3] clusterProfiler_3.18.1      clusterExperiment_2.11.2
##  [5] tradeSeq_1.4.0              magrittr_2.0.1
##  [7] monocle3_1.0.0              SingleCellExperiment_1.12.0
##  [9] SummarizedExperiment_1.20.0 GenomicRanges_1.42.0
## [11] GenomeInfoDb_1.26.7         IRanges_2.24.1
## [13] S4Vectors_0.28.1            MatrixGenerics_1.2.1
## [15] matrixStats_0.61.0          Biobase_2.50.0
## [17] BiocGenerics_0.36.1         circlize_0.4.13
## [19] RColorBrewer_1.1-2          dplyr_1.0.7
## [21] ggplot2_3.3.5               ComplexHeatmap_2.6.2
## [23] SeuratObject_4.0.4          Seurat_4.0.5
##
## loaded via a namespace (and not attached):
##   [1] rsvd_1.0.5              ica_1.0-2               zinbwave_1.12.0
##   [4] class_7.3-17            foreach_1.5.1           lmtest_0.9-39
##   [7] crayon_1.4.2            spatstat.core_2.3-2     MASS_7.3-53
##  [10] rhdf5filters_1.2.1      nlme_3.1-153            qlcMatrix_0.9.7
##  [13] GOSemSim_2.16.1         rlang_0.4.12            XVector_0.30.0
##  [16] ROCR_1.0-11             irlba_2.3.5             limma_3.46.0
##  [19] phylobase_0.8.10        BiocParallel_1.24.1     rjson_0.2.20
##  [22] bit64_4.0.5             glue_1.5.1              pheatmap_1.0.12
##  [25] rngtools_1.5.2          sctransform_0.3.2       spatstat.sparse_2
    .0-0
##  [28] classInt_0.4-3          DOSE_3.16.0             spatstat.geom_2.3-0
##  [31] VGAM_1.1-5              tidyselect_1.1.1        fitdistrplus_1.1-6
##  [34] XML_3.99-0.8            tidyr_1.1.4             zoo_1.8-9
##  [37] sf_1.0-4                xtable_1.8-4            spData_2.0.1
##  [40] evaluate_0.14           cli_3.1.0               zlibbioc_1.36.0
##  [43] rstudioapi_0.13         miniUI_0.1.1.1          sp_1.4-6
##  [46] rpart_4.1-15            fastmatch_1.1-3         pbmcapply_1.5.0
##  [49] locfdr_1.1-8            shiny_1.7.1             BiocSingular_1.6.0
##  [52] xfun_0.28               clue_0.3-60             cluster_2.1.0
##  [55] tidygraph_1.2.0         tibble_3.1.6            expm_0.999-6
##  [58] ggrepel_0.9.1           ape_5.5                 listenv_0.8.0
##  [61] png_0.1-7               future_1.23.0           withr_2.4.3
##  [64] bitops_1.0-7            slam_0.1-49             ggforce_0.3.3
##  [67] plyr_1.8.6              sparsesvd_0.2           e1071_1.7-9
##  [70] coda_0.19-4             pillar_1.6.4            GlobalOptions_0.1.2
##  [73] cachem_1.0.6            kernlab_0.9-29          raster_3.5-2
##  [76] GetoptLong_1.0.5        gmodels_2.18.1          vctrs_0.3.8
##  [79] ellipsis_0.3.2          generics_0.1.1          NMF_0.23.0
##  [82] tools_4.0.3             rncl_0.8.4              munsell_0.5.0
##  [85] tweenr_1.0.2            fgsea_1.16.0            proxy_0.4-26
##  [88] DelayedArray_0.16.3     fastmap_1.1.0           HSMMSingleCell_1
    .10.0
##  [91] compiler_4.0.3          abind_1.4-5             httpuv_1.6.3
##  [94] pkgmaker_0.32.2         plotly_4.10.0           GenomeInfoDbData_1
    .2.4
##  [97] gridExtra_2.3           edgeR_3.32.1            lattice_0.20-41
## [100] deldir_1.0-6            utf8_1.2.2              later_1.3.0
## [103] wk_0.5.0                jsonlite_1.7.2          scales_1.1.1
## [106] princurve_2.1.6         docopt_0.7.1            pbapply_1.5-0
```

```
## [109] genefilter_1.72.1      lazyeval_0.2.2          LearnBayes_2.15.1   72
## [112] promises_1.2.0.1       doParallel_1.0.16       goftest_1.2-3       73
## [115] spatstat.utils_2.2-0   reticulate_1.22         rmarkdown_2.11      74
## [118] cowplot_1.1.1          textshaping_0.3.6       Rtsne_0.15          75
## [121] downloader_0.4         softImpute_1.4-1        uwot_0.1.11         76
## [124] igraph_1.2.9           HDF5Array_1.18.1        survival_3.2-7      77
## [127] yaml_2.2.1             systemfonts_1.0.3       DDRTree_0.1.5       78
## [130] htmltools_0.5.2        memoise_2.0.1           locfit_1.5-9.4      79
## [133] graphlayouts_0.7.2     viridisLite_0.4.0       digest_0.6.29       80
## [136] assertthat_0.2.1       mime_0.12               densityClust_0.3    81
## [139] registry_0.5-1         units_0.7-2             RSQLite_2.2.9       82
## [142] yulab.utils_0.0.4      future.apply_1.8.1      data.table_1.14.2   83
## [145] blob_1.2.2             RNeXML_2.4.5            ragg_1.2.1          84
## [148] fastICA_1.2-3          splines_4.0.3           labeling_0.4.2      85
## [151] Rhdf5lib_1.12.1        Cairo_1.5-12.2          RCurl_1.98-1.5      86
## [154] monocle_2.18.0         hms_1.1.1               rhdf5_2.34.0        87
## [157] colorspace_2.0-2       BiocManager_1.30.16     shape_1.4.6         88
## [160] Rcpp_1.0.7             RANN_2.6.1              enrichplot_1.10.2   89
## [163] fansi_0.5.0            parallelly_1.29.0       R6_2.5.1            90
## [166] ggridges_0.5.3         lifecycle_1.0.1         gdata_2.18.0        91
## [169] leiden_0.3.9           DO.db_2.9               Matrix_1.3-4        92
## [172] howmany_0.3-1          qvalue_2.22.0           RcppAnnoy_0.0.19    93
## [175] iterators_1.0.13       stringr_1.4.0           htmlwidgets_1.5.4   94
## [178] beachmat_2.6.4         polyclip_1.10-0         purrr_0.3.4         95
## [181] shadowtext_0.0.9       terra_1.4-22            mgcv_1.8-33         96
## [184] globals_0.14.0         patchwork_1.1.1         slingshot_1.8.0     97
## [187] codetools_0.2-18       GO.db_3.12.1            FNN_1.1.3           98
## [190] gtools_3.9.2           prettyunits_1.1.1       gridBase_0.4-7      99
## [193] gtable_0.3.0           DBI_1.1.1               ggfun_0.0.4         100
## [196] tensor_1.5             httr_1.4.2              highr_0.9           101
## [199] KernSmooth_2.23-20     stringi_1.7.6           progress_1.2.2      102
## [202] reshape2_1.4.4         farver_2.1.0            uuid_1.0-3          103
## [205] spdep_1.1-12           annotate_1.68.0         viridis_0.6.2       104
## [208] magick_2.7.3           xml2_1.3.3              combinat_0.0-8      105
## [211] rvcheck_0.2.1          boot_1.3-25             s2_1.0.7            106
## [214] ade4_1.7-18            scattermore_0.7         bit_4.0.4           107
## [217] scatterpie_0.1.7       spatstat.data_2.1-0     ggraph_2.0.5        108
## [220] pkgconfig_2.0.3        knitr_1.36                                  109
```

9 References