

Mafb-restricted local monocyte proliferation precedes lung interstitial macrophage differentiation

10-DE genes across pseudotime

2022-03-09 17:14:53 +0100

Abstract

Resident tissue macrophages (RTM) are differentiated immune cells populating distinct niches and exhibiting important tissue-supportive functions. RTM maintenance is thought to rely on either monocyte engraftment and differentiation, or RTM self-renewal. Here, we developed an inducible mouse model of lung interstitial macrophage (IM) niche depletion and repopulation to investigate IM development *in vivo*. Using time-course single-cell RNA-sequencing analyses, bone marrow chimeras and gene targeting, we found that engrafted Ly6C+ classical monocytes could self-renew locally in a CSF1R-dependent manner before their differentiation into RTM. We further showed that the switch from monocyte proliferation towards IM subset specification was controlled by MafB, while c-Maf specifically regulated the identity of the CD206+ IM subset. Our data shed new light on the transcriptional regulation of IM development and provide evidence that, in the mononuclear phagocyte system, self-renewal is not merely restricted to myeloid progenitor cells and mature macrophages, but is also a tightly regulated capability of mature monocytes developing into RTM *in vivo*.

Contents

1 Description	3
2 Prepare data	3
3 DE gene expression across IM-differentiation	4
3.1 Annotate the cells associated to either differentiation of CD206+ IMs or CD206- IMs	7
4 TradeSeq analysis for the differentiation of monocytes to either of IM subsets	7
4.1 Construct sce object for TradeSeq	7
4.2 Clustering using RSEC, clusterExperiment	9
5 Show gene expression pattern calculated by TradeSeq in heatmap	11
5.1 Data preparation	11
5.2 Draw heatmap	11
5.3 Annotate DE genes as CD206+/CD206- IM differentiation specific or common genes	12
5.4 Draw heatmap with expression patterns of unchanged/common genes in the order of pseudotime	15
5.5 Make with changed/specific genes	17
5.6 GO enrichment on the changed/specific genes	18
6 Functionality analysis of DE genes across pseudotime (common genes)	20
6.1 Manually classify genes by expression timing peak	20
6.2 GO/KEGG enrichment analysis with 3 classes of common genes	21
7 Show gene expression pattern with TradeSeq results	26
7.1 Class 1 common genes	26
7.2 Class 2 common genes	28
7.3 Class 3 common genes	29

8 Session information	45
References	48

1 Description

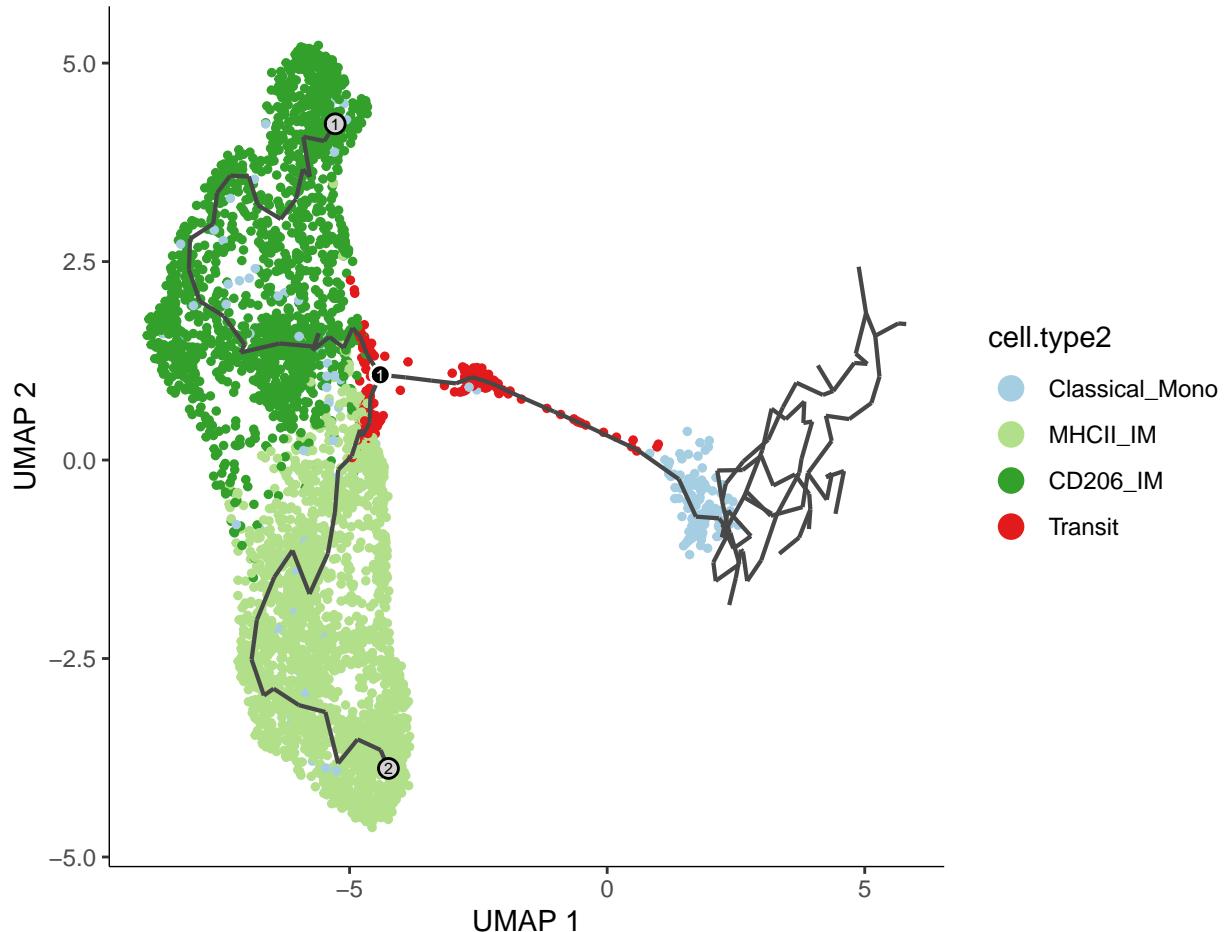
To compare the expression patterns of DE genes across pseudotime, the counts matrix, pseudotime and cell weights calculated above were then used as input in fitGAM function (TradeSeq package).¹ The association of average expression of each gene with pseudotime was tested using associationTest and the DE genes between CD206+ and CD206- IM trajectories were calculated with diffEndTest function. The value of the estimated smoother on a grid of pseudotimes was estimated for each of DE gene using predictSmooth. The DE genes with waldStat > 70 and |logFC| > 2 were annotated as “changed genes”, meaning that their expression patterns were different in CD206+ and CD206- IM trajectories, while the rest of DE genes were considered as “unchanged genes”, meaning that the expression patterns were similar in both trajectories. Finally, the scaled estimated smoothers calculated by predictSmooth were used to build heatmap with ComplexHeatmap package.²

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")
```

Show trajectory in UMAP plot:

```
pal2 <- c(`Classical_Mono`="#A6CEE3",  
           `MHCII_IM`="#B2DF8A",  
           `CD206_IM`="#33A02C",  
           `Transit` = "#E31A1C")  
  
plot_cells(cds = mo, color_cells_by = "cell.type2",  
           cell_size = 1, label_cell_groups = FALSE,  
           label_branch_points = TRUE, label_leaves = TRUE,  
           label_roots = FALSE, alpha = 1) +  
           scale_color_manual(values = pal2)
```



```
ggsave(filename = ".../Figures/UMAPplot_trajectory.pdf", width = 6.5, height = 5)
```

3 DE gene expression across IM-differentiation

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

Prepare matrix with z-scores, smoothed 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
```

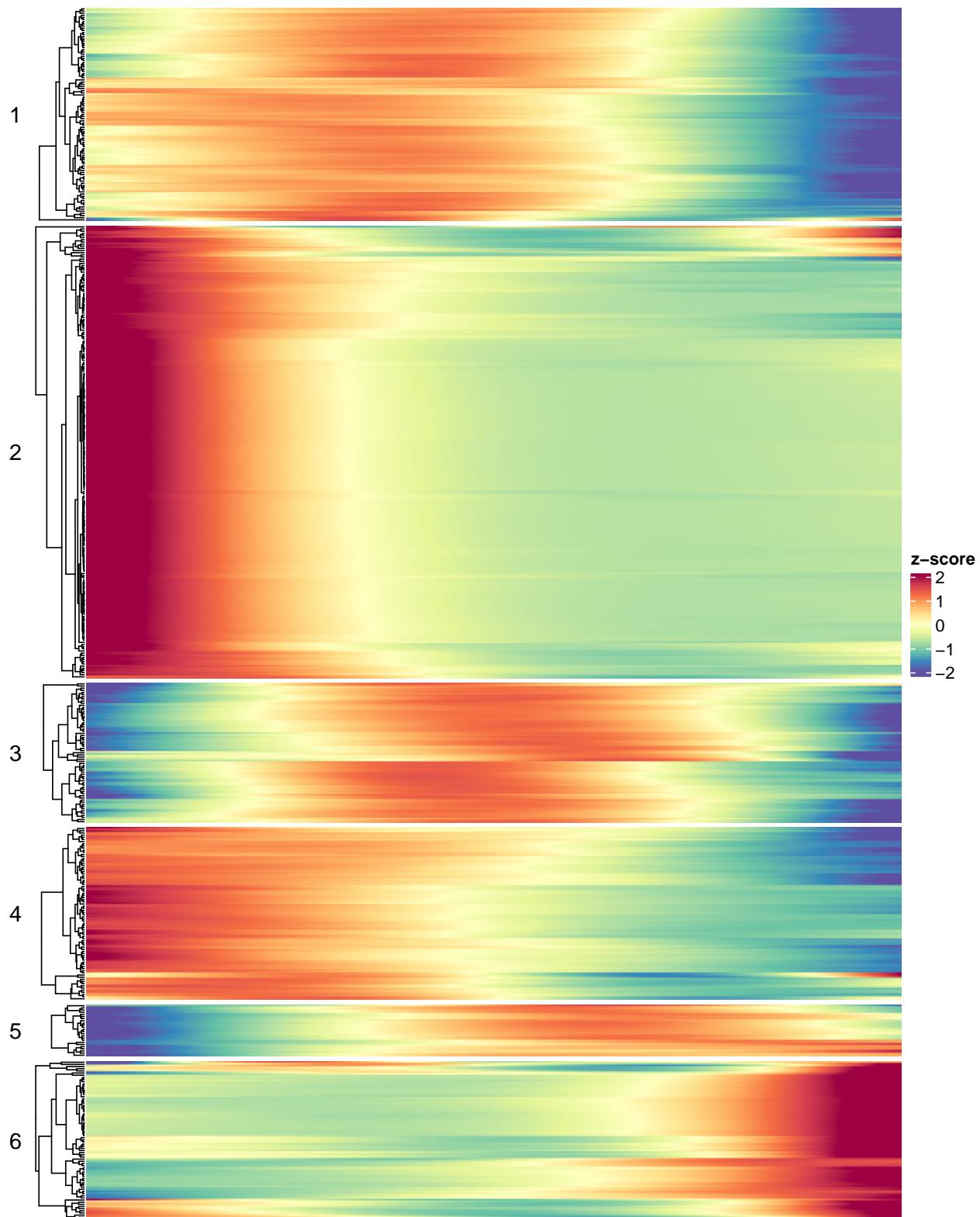
Show DE genes in unsupervised heatmap.

```
#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)                                     1
# Get the closest vertice for every cell           2
y_to_cells <- mo@principal_graph_aux$UMAP$pr_graph_cell_proj_closest_ 3
  vertex%>%as.data.frame()

y_to_cells$cells <- rownames(y_to_cells)           4
y_to_cells$Y <- y_to_cells$V1                      5

# Get the root vertices                           6
# It is the same node as above                  7
root <- mo@principal_graph_aux$UMAP$root_pr_nodes 8

principalgraph <- mo@principal_graph$UMAP          9

# Get the other endpoints                         10
endpoints <- names(which(igraph::degree(principalgraph) == 1))    11
endpoints <- endpoints[!endpoints %in% root]        12

# For each endpoint                            13
cellWeights <- lapply(endpoints, function(endpoint) { 14
  # We find the path between the endpoint and the root 15
  path <- igraph::shortest_paths(principalgraph, root, endpoint)$vpath 16
  [[1]]
  path <- as.character(path)                         17
  # We find the cells that map along that path      18
  df <- y_to_cells[y_to_cells$Y %in% path, ]         19
  df <- data.frame(weights = as.numeric(colnames(mo) %in% df$cells)) 20
  colnames(df) <- endpoint                         21
  return(df)
}) %>% do.call(what = 'cbind', args = .) %>%
  as.matrix()                                       22
rownames(cellWeights) <- colnames(mo)               23
colnames(cellWeights) <- c("CD206_IM_branch", "MHCII_IM_branch") 24
pseudotime <- matrix(mo@principal_graph_aux$UMAP$pseudotime, ncol = ncol( 25
  cellWeights),                                26
  nrow = ncol(mo), byrow = FALSE)                27
rownames(pseudotime) <- colnames(mo)               28

```

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

4.1 Construct sce object for TradeSeq

```

suppressMessages(library(tradeSeq))                 1

# this step is VERY time consuming                2
sce <- fitGAM(counts = mo@assays$data$counts,      3
  pseudotime = pseudotime,
  cellWeights = cellWeights)                     4

```

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

5
6

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)
```

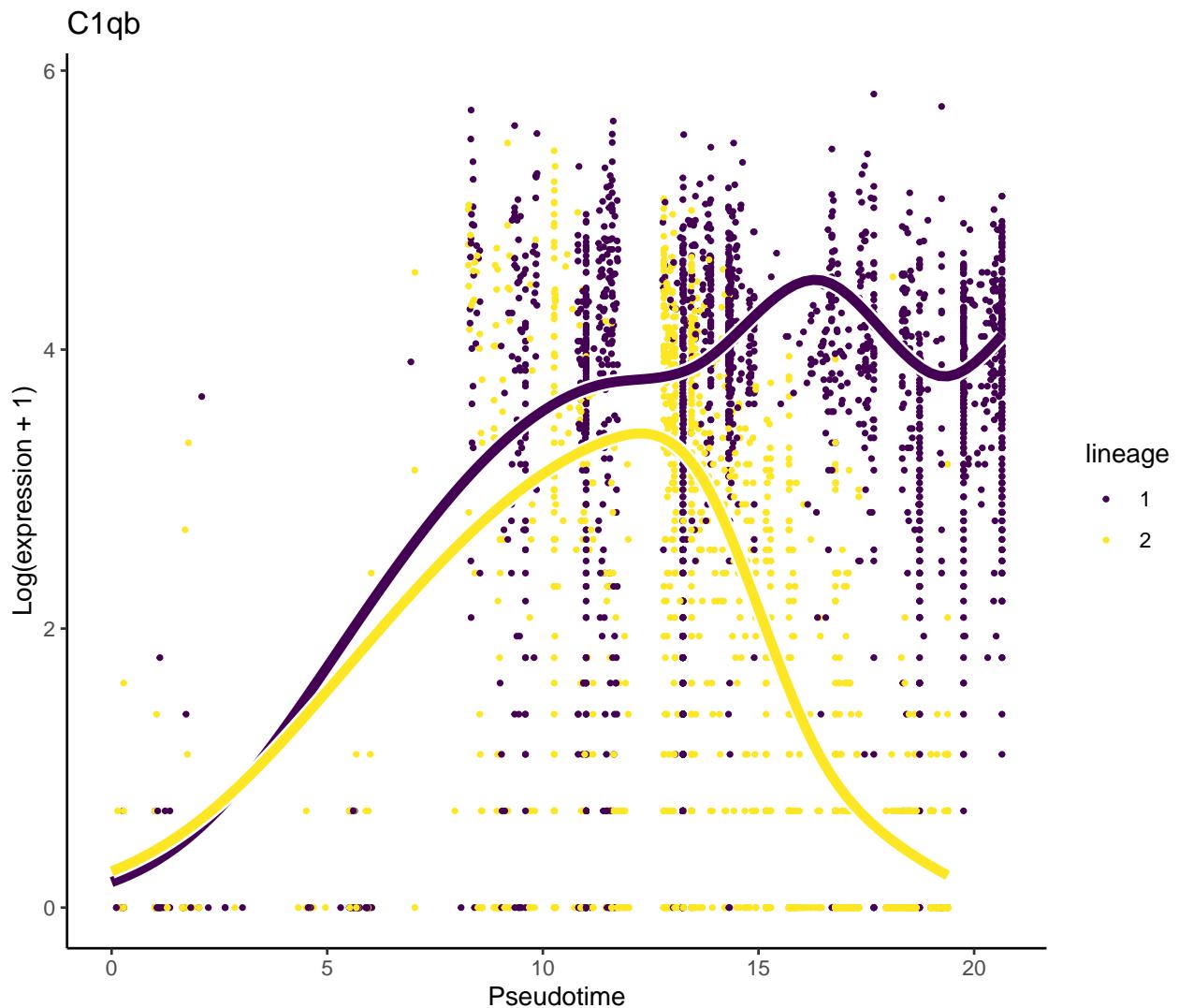
```
## # A tibble: 6 x 4
##   waldStat     df      pvalue  meanLogFC
##       <dbl>    <dbl>      <dbl>      <dbl>
## 1     210.      9     0.222
## 2     28.4      9    0.000815
## 3       NA      NA     NA     0.121
## 4     41.8      9    0.00000360
## 5     36.4      9    0.0000330
## 6     45.1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

1
2
3
4
5
6
7
8
9

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)
```

1
2
3
4
5
6



What's the top genes?

```

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

## [1] "C1qb"      "Ctsb"       "C1qa"       "Selenop"    "Csf1r"      "Timp2"      "Pf4"        1  

## [8] "C1qc"      "Serinc3"    "Cd209a"    "Lsp1"       "Lgmn"       "Apoe"       "Blvrb"     2  

## [15] "Olfm1"     "Tnip3"      "Rpl13"      "Ninj1"      "Rpl28"      "H2-DMb1"   3

```

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
## Running Clustering on Parameter Combinations...
## done.

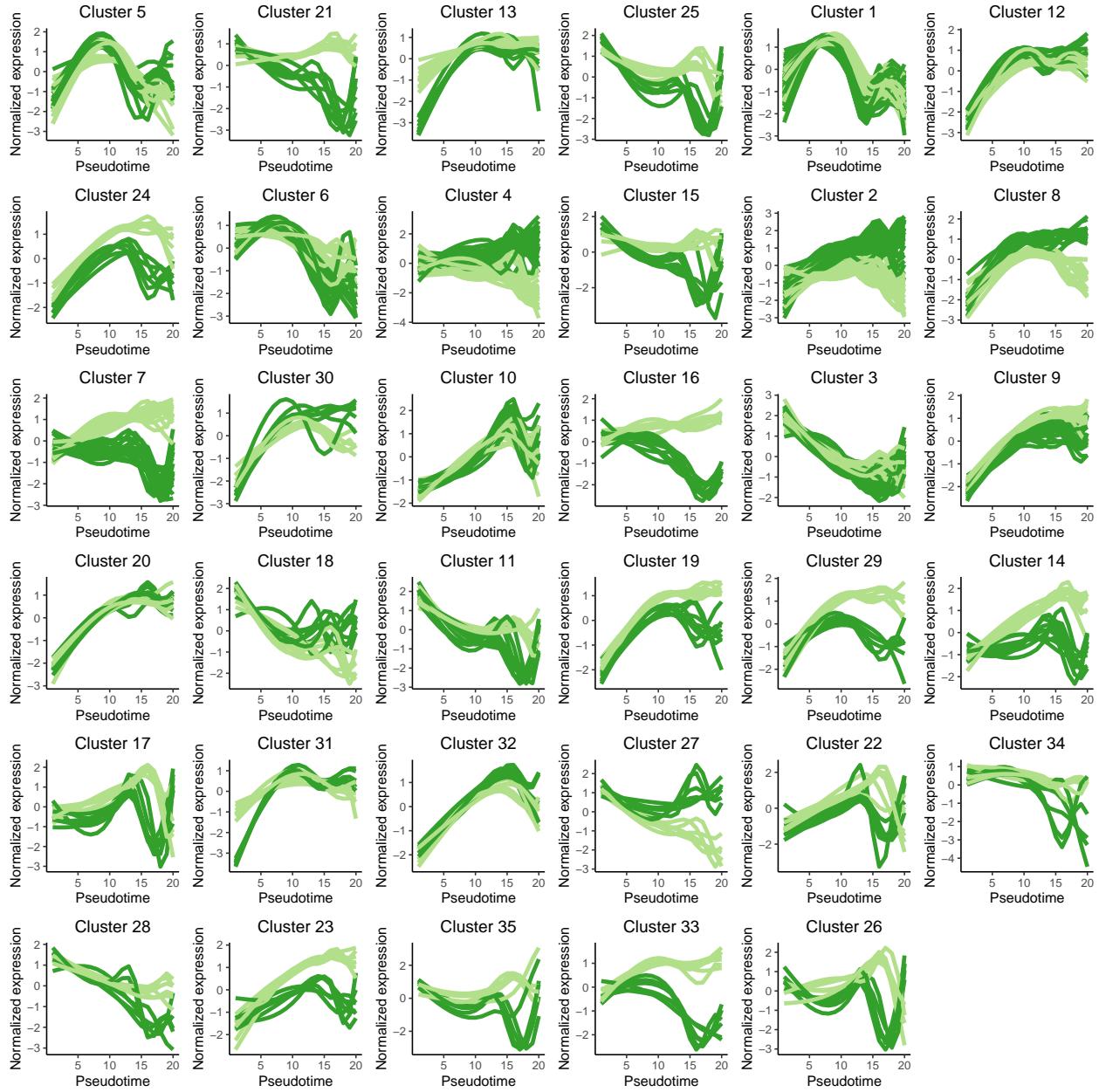
clusterLabels <- primaryCluster(clusPat$rsec)

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

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.

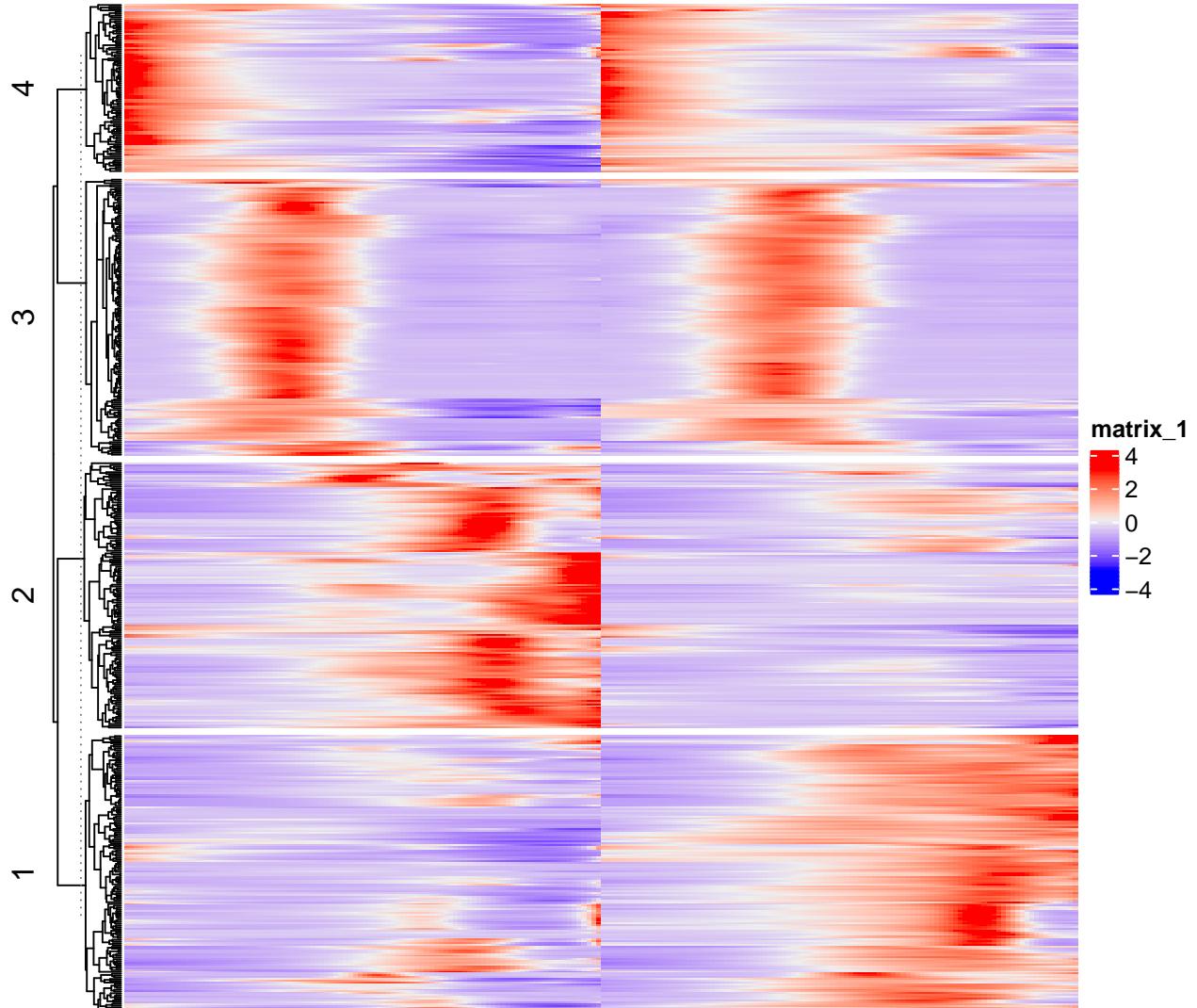
```
yhatSmooth <- predictSmooth(sce, gene = genes, nPoints = 100, tidy = FALSE) 1
      )
yhatSmoothScaled <- t(scale(t(yhatSmooth))) 2
```

5.2 Draw heatmap

```

heatSmooth <- Heatmap(yhatSmoothScaled, cluster_columns = FALSE, show_row_ 1
  names = FALSE, show_column_names = FALSE, row_km = 4)
heatSmooth <- draw(heatSmooth) 2

```



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) 1
 2

```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	1
##	0.0027	3.8599	41.4571	123.7353	168.7434	1263.9587	2

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")  

1  

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")  

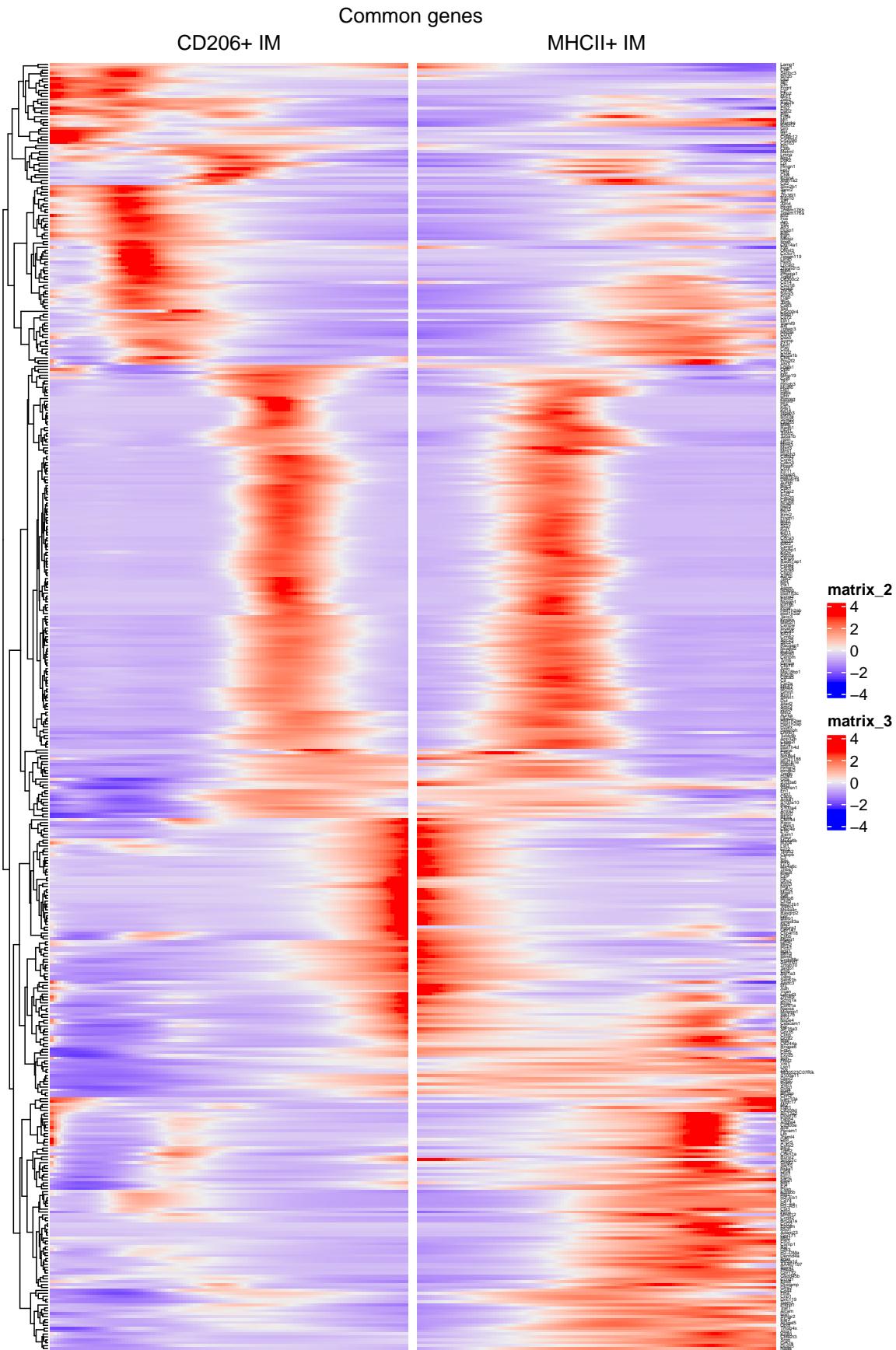
2  

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

3  

4  

5
```



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

Let's find the expression peak of each gene:

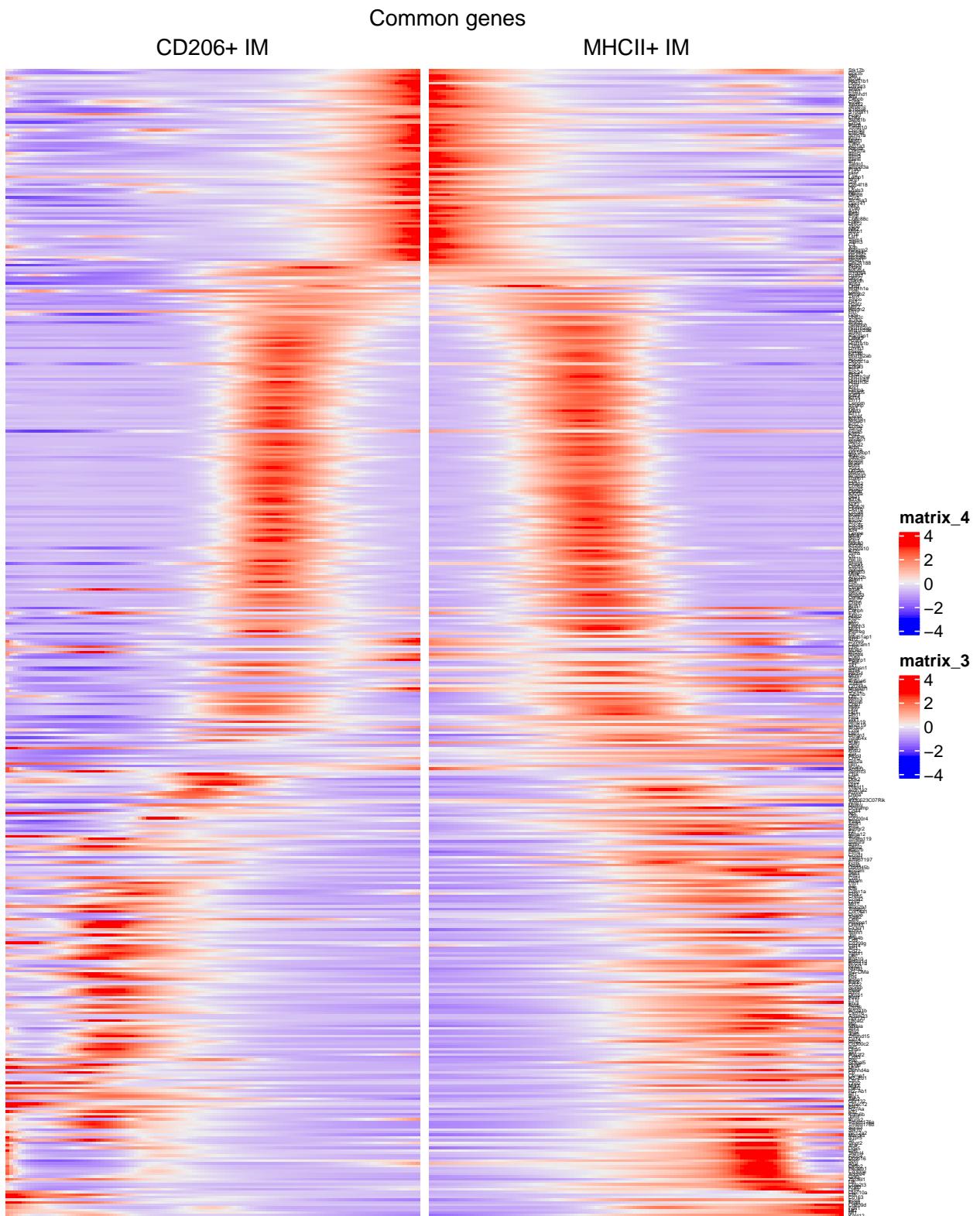
```

1  orderbyExpressionPeak <- function(x, # matrix
2      decreasing = FALSE,
3      output.position = FALSE # if true, give
4      relative position 0 - 1, or output
5      order.
6      )
7
8  indx.peak <- apply(x, 1 , which.max)
9
10 if(output.position) {
11   po <- indx.peak/nrow(x)
12   if (! length(rownames(x)) == 0) {names(po) <- rownames(x)}
13   return(po)
14 } else {
15   o <- order(indx.peak)
16   if (! length(rownames(x)) == 0) {names(o) <- rownames(x)[o]}
17   return (o)
18 }
```

Make average peak pseudotime peak for each gene:

```

1  po.cd206 <- orderbyExpressionPeak(yhatSmoothScaled[genes.noChange , 1:100] ,
2      output.position = TRUE)
3  po.mhcii <- orderbyExpressionPeak(yhatSmoothScaled[genes.noChange ,
4      101:200] , output.position = TRUE)
5  order.mean <- order ( ( po.cd206 + po.mhcii ) /2)
6
7  heatSmooth_cd206.unchanged.ordered <- Heatmap(yhatSmoothScaled[genes.
8      noChange , 100:1] , cluster_columns = FALSE, show_row_names = FALSE,
9      show_column_names = FALSE, row_order = order.mean, column_title =
10      "CD206+IM")
11
12  heatSmooth_combined.unchanged.ordered <- draw ( heatSmooth_cd206.unchanged
13      .ordered + heatSmooth_mhcii.unchanged, column_title = "Common genes",
14      auto_adjust = FALSE)
```



```

pdf(file = ".../Figures/Heatmap_common_genes_IMs_diff_across_pseudotime.pdf" 1
  , width = 8, height = 10)
heatSmooth_combined.unchanged.ordered
dev.off() 2
 3

```

5.5 Make with changed/specific genes

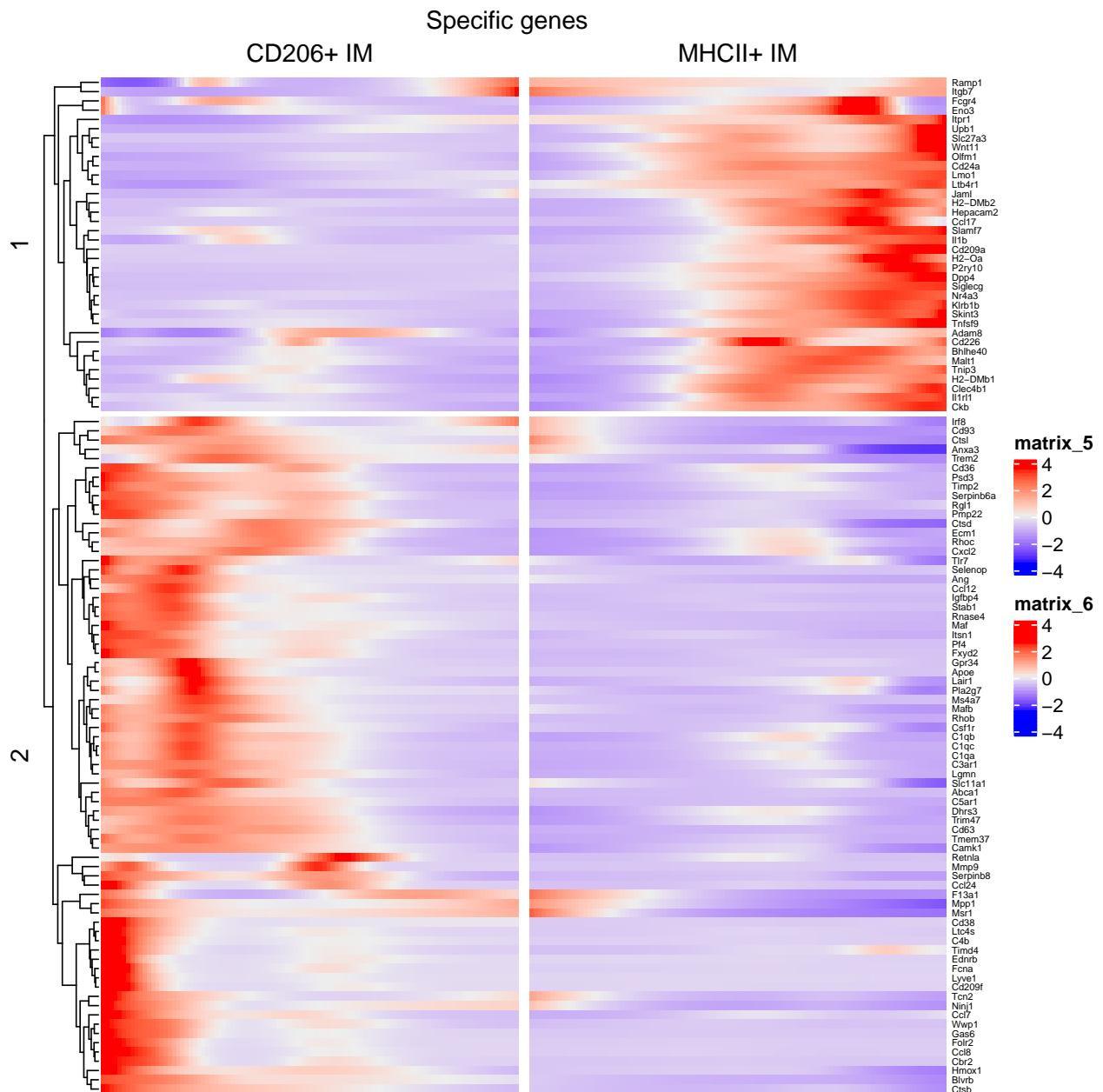
```

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 = 5), show_column_names = FALSE, column_title = "MHCII+
IM")

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

```



```

pdf(file = "../Figures/Heatmap_specific_CD206_IM_genes_IMs_diff_across_
    pseudotime.pdf", width = 8, height = 8)
heatSmooth_combined.changed
dev.off()

geneSymbol.changed.mhcii <- read.csv("./geneSymbol_changed_mhcii_IM.csv",
    col.names = FALSE)[[1]]
geneSymbol.changed.cd206 <- read.csv("./geneSymbol_changed_cd206_IM.csv",
    col.names = FALSE)[[1]]

```

MHCII IM high genes:

geneSymbol.changed.mhcii	1
## [1] "Ramp1" "Itgb7" "Fcgr4" "Eno3" "Itpr1" "Upb1"	1
## [7] "Slc27a3" "Wnt11" "Olfm1" "Cd24a" "Lmo1" "Ltb4r1"	2
## [13] "Jaml" "H2-DMb2" "Hepacam2" "Ccl17" "Slamf7" "Il1b"	3
## [19] "Cd209a" "H2-Oa" "P2ry10" "Dpp4" "Siglecg" "Nr4a3"	4
## [25] "Klrb1b" "Skint3" "Tnfsf9" "Adam8" "Cd226" "Bhlhe40"	5
## [31] "Malt1" "Tnip3" "H2-DMb1" "Clec4b1" "Il1rl1" "Ckb"	6

geneSymbol.changed.cd206	1
## [1] "Irf8" "Cd93" "Ctsl" "Anxa3" "Trem2" "Cd36"	1
## [7] "Psd3" "Timp2" "Serpineb6a" "Rgl1" "Pmp22" "Ctsd"	2
## [13] "Ecm1" "Rhoc" "Cxcl2" "Tlr7" "Selenop" "Ang"	3
## [19] "Ccl12" "Igfbp4" "Stab1" "Rnase4" "Maf" "Itsn1"	4
"	
## [25] "Pf4" "Fxyd2" "Gpr34" "Apoe" "Lair1" "	5
Pla2g7"	
## [31] "Ms4a7" "Mafb" "Rhob" "Csf1r" "C1qb" "C1qc"	6
## [37] "C1qqa" "C3ar1" "Lgmn" "Slc11a1" "Abca1" "C5ar1"	7
"	
## [43] "Dhrs3" "Trim47" "Cd63" "Tmem37" "Camk1" "	8
Retnla"	
## [49] "Mmp9" "Serpineb8" "Ccl24" "F13a1" "Mpp1" "Msrl"	9
## [55] "Cd38" "Ltc4s" "C4b" "Timd4" "Ednrb" "Fcna"	10
## [61] "Lyve1" "Cd209f" "Tcn2" "Ninj1" "Ccl7" "Wwp1"	11
## [67] "Gas6" "Folr2" "Ccl8" "Cbr2" "Hmox1" "Blvrb"	12
"	
## [73] "Ctsb"	13

write.csv(geneSymbol.changed.cd206, file = "./geneSymbol_changed_cd206_IM. csv", quote = FALSE, row.names = FALSE)	1
write.csv(geneSymbol.changed.mhcii, file = "./geneSymbol_changed_mhcii_IM. csv", quote = FALSE, row.names = FALSE)	2

5.6 GO enrichment on the changed/specific genes

suppressMessages(library(clusterProfiler))	1
source("../R/entrez2symbol.R")	2
source("../R/replaceEntrezID.R")	3

5.6.1 GO enrichment on MHCII IM

```

symb <- geneSymbol.changed.mhcii
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
  , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
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_"
  # MHCII_IM_specific_genes.csv")
result.enrichGO@result

```

	## # A tibble: 1,257 x 9	1	
	## ID Description GeneRatio BgRatio pvalue p.adjust qvalue	2	
	geneID Count	3	
	<chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <	4	
	chr> <int>	5	
## 1	GO:00508~ regulation~ 9/36	321/23~ 1.07e-9 1.35e-6 8.17e-7	6
	Cd24a~ 9		
## 2	GO:00026~ positive r~ 9/36	479/23~ 3.46e-8 1.17e-5 7.11e-6	7
	Cd24a~ 9		
## 3	GO:19030~ positive r~ 7/36	221/23~ 4.12e-8 1.17e-5 7.11e-6	8
	Cd24a~ 7		
## 4	GO:00071~ leukocyte ~ 8/36	345/23~ 4.45e-8 1.17e-5 7.11e-6	9
	Itgb7~ 8		
## 5	GO:00508~ positive r~ 9/36	496/23~ 4.67e-8 1.17e-5 7.11e-6	10
	Cd24a~ 9		
## 6	GO:00224~ positive r~ 7/36	265/23~ 1.42e-7 2.98e-5 1.80e-5	11
	Cd24a~ 7		
## 7	GO:00026~ regulation~ 8/36	440/23~ 2.86e-7 4.76e-5 2.88e-5	12
	Cd24a~ 8		
## 8	GO:00018~ positive r~ 8/36	449/23~ 3.34e-7 4.76e-5 2.88e-5	13
	Wnt11~ 8		
## 9	GO:19030~ regulation~ 7/36	309/23~ 4.01e-7 4.76e-5 2.88e-5	14
	Cd24a~ 7		
## 10	GO:00456~ regulation~ 6/36	192/23~ 4.56e-7 4.76e-5 2.88e-5	
	Cd24a~ 6		
## # ... with 1,247 more rows			

5.6.2 GO enrichment on CD206+ IM

```

symb <- geneSymbol.changed.cd206
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
  , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
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_"
  # CD206_IM_specific_genes.csv")
result.enrichGO@result

```

	## # A tibble: 1,842 x 9	1
	## ID Description GeneRatio BgRatio pvalue p.adjust qvalue	2
	geneID Count	3
	<chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <	4
	chr> <int>	5

## 1	GO:005~ leukocyte ~	17/70	360/23~	4.11e-16	7.56e-13	4.90e-13	4
Trem2~	17						
## 2	GO:003~ leukocyte ~	14/70	219/23~	3.30e-15	2.03e-12	1.31e-12	5
Cxcl2~	14						
## 3	GO:009~ myeloid le~	14/70	219/23~	3.30e-15	2.03e-12	1.31e-12	6
Trem2~	14						
## 4	GO:006~ cell chemo~	14/70	303/23~	2.90e-13	1.34e-10	8.65e-11	7
Cxcl2~	14						
## 5	GO:000~ regulation~	12/70	217/23~	2.05e-12	7.55e-10	4.89e-10	8
Trem2~	12						
## 6	GO:007~ granulocyt~	10/70	126/23~	4.44e-12	1.36e- 9	8.82e-10	9
Cxcl2~	10						
## 7	GO:007~ mononuclea~	9/70	89/233~	6.08e-12	1.60e- 9	1.04e- 9	10
Ccl12~	9						
## 8	GO:003~ neutrophil~	9/70	99/233~	1.62e-11	3.72e- 9	2.41e- 9	11
Cxcl2~	9						
## 9	GO:009~ granulocyt~	10/70	155/23~	3.53e-11	6.93e- 9	4.48e- 9	12
Cxcl2~	10						
## 10	GO:000~ positive r~	10/70	156/23~	3.76e-11	6.93e- 9	4.48e- 9	13
Trem2~	10						
## # ... with 1,832 more rows							14

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

```
common.genes <- heatSmooth_cd206.unchanged.ordered@row_names_param$labels[ 1
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
2
```

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

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
2
```

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

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
2
```

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

Save gene lists:

```
write.csv(genes.class1, file = "./common_genes_class1.csv", quote = FALSE) 1
write.csv(genes.class2, file = "./common_genes_class2.csv", quote = FALSE) 2
write.csv(genes.class3, file = "./common_genes_class3.csv", quote = FALSE) 3
```

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

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") 4
# write.csv(result.enrichKEGG@result, file = "./Results_enrichment/
  enrichKEGG_common_genes_class1.csv")
result.enrichKEGG@result
```

```
## # A tibble: 149 x 9
##       ID      Description  GeneRatio BgRatio    pvalue p.adjust    qvalue
##   <chr>    <chr>     <chr>     <chr>     <dbl>     <dbl>     <dbl>    <
## 1 mmu04145 Phagosome    7/36     182/89~  6.50e-6  0.000969  8.96e-4
## 2 Thbs1~        7
## 3 mmu05152 Tuberculosis 6/36     180/89~  7.19e-5  0.00535   4.95e-3
## 4 Cebpb~        6
## 5 mmu05140 Leishmanias~ 3/36     70/8949  2.72e-3  0.119    1.10e-1
## 6 Cybb/~        3
## 7 mmu04918 Thyroid hor~ 3/36     74/8949  3.19e-3  0.119    1.10e-1
## 8 Plcb1~        3
## 9 mmu04970 Salivary se~ 3/36     86/8949  4.87e-3  0.145    1.34e-1
## 10 Plcb1~       3
## 11 mmu04610 Complement ~ 3/36     93/8949  6.05e-3  0.150    1.39e-1
## 12 Plaur~        3
## 13 mmu04613 Neutrophil ~ 4/36     207/89~  9.18e-3  0.168    1.55e-1
## 14 Plcb1~       4
## 15 mmu04960 Aldosterone~ 2/36     38/8949  1.01e-2  0.168    1.55e-1
## 16 Scnn1~        2
## 17 mmu04621 NOD-like re~ 4/36     213/89~  1.01e-2  0.168    1.55e-1
## 18 Ifi20~       4
## 19 mmu04973 Carbohydrat~ 2/36     48/8949  1.58e-2  0.235    2.18e-1
## 20 Plcb1~       2
## # ... with 139 more rows
```

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
##   ID             Description GeneRatio BgRatio    pvalue p.adjust    qvalue
##   <chr>          <chr>     <chr>     <chr>     <dbl>    <dbl>    <dbl>    <
##   geneID Count
##   <chr> <int>
##   chr> <int>
##   1 GO:00975~ myeloid le~ 10/72      219/23~ 1.40e-9 2.16e-6 1.65e-6
##   Gpr35~ 10
##   2 GO:00451~ cellular e~ 7/72       72/233~ 2.48e-9 2.16e-6 1.65e-6
##   Sell/~ 7
##   3 GO:00509~ leukocyte ~ 11/72      360/23~ 1.32e-8 7.68e-6 5.86e-6
##   Gpr35~ 11
##   4 GO:00305~ leukocyte ~ 9/72       219/23~ 2.45e-8 1.07e-5 8.14e-6
##   Gpr35~ 9
##   5 GO:00321~ positive r~ 11/72      418/23~ 6.08e-8 2.12e-5 1.61e-5
##   Sell/~ 11
##   6 GO:00603~ type I int~ 5/72       40/233~ 1.47e-7 3.64e-5 2.78e-5
##   Samhd~ 5
##   7 GO:00713~ cellular r~ 5/72       40/233~ 1.47e-7 3.64e-5 2.78e-5
##   Samhd~ 5
##   8 GO:00427~ defense re~ 11/72      464/23~ 1.74e-7 3.64e-5 2.78e-5
##   Slpi/~ 11
##   9 GO:00313~ positive r~ 9/72       278/23~ 1.88e-7 3.64e-5 2.78e-5
##   Ifi20~ 9
##   10 GO:00343~ response t~ 5/72      45/233~ 2.70e-7 4.70e-5 3.59e-5
##   Samhd~ 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" 1
  , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
result.enrichKEGG <- enrichKEGG(de_entrez, organism = "mmu", keyType = "2
  ncbi-geneid")
result.enrichKEGG <- replaceEntrezID(result.enrichKEGG, organism = "mmu") 3
# write.csv(result.enrichKEGG@result, file = "./Results_enrichment/
  enrichKEGG_common_genes_class2.csv")
result.enrichKEGG@result                                         4

```

```

## # A tibble: 88 x 9
##   ID             Description GeneRatio BgRatio    pvalue p.adjust    qvalue
##   <chr>          <chr>     <chr>     <chr>     <dbl>    <dbl>    <dbl>    <
##   geneID Count
##   <chr> <int>
##   chr> <int>
##   1 mmu04114 Oocyte mei~ 11/50      121/89~ 4.17e-11 2.62e-9 2.26e-9
##   Aurka~ 11
##   2 mmu04110 Cell cycle 11/50      125/89~ 5.96e-11 2.62e-9 2.26e-9
##   Ccnb1~ 11

```

## 3 mmu04914 Progesterone~	9/50	92/8949	1.53e- 9	4.48e-8	3.86e-8	6
Aurka~	9					
## 4 mmu04115 p53 signal~	5/50	72/8949	4.68e- 5	1.03e-3	8.87e-4	7
Ccnb1~	5					
## 5 mmu00240 Pyrimidine~	4/50	56/8949	2.56e- 4	4.50e-3	3.88e-3	8
Rrm2/~	4					
## 6 mmu04218 Cellular s~	6/50	184/89~	5.22e- 4	7.65e-3	6.59e-3	9
Ccnb1~	6					
## 7 mmu05222 Small cell~	4/50	93/8949	1.75e- 3	2.20e-2	1.89e-2	Fn1 10
/C~	4					
## 8 mmu05166 Human T-ce~	6/50	250/89~	2.54e- 3	2.64e-2	2.27e-2	11
Bub1b~	6					
## 9 mmu05132 Salmonella~	6/50	253/89~	2.70e- 3	2.64e-2	2.27e-2	12
Gapdh~	6					
## 10 mmu04512 ECM-recept~	3/50	88/8949	1.29e- 2	1.14e-1	9.79e-2	Fn1 13
/H~	3					
## # ... with 78 more rows						14

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									1
## ID Description GeneRatio BgRatio				pvalue	p.adjust	qvalue			2
geneID Count									3
## <chr> <chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	4
## <chr> <int>									5
## 1 GO:000~ chromosome~	50/127	324/23~	7.39e-60	1.32e-56	1.08e-56				6
Ube2c~	50								7
## 2 GO:009~ nuclear ch~	43/127	262/23~	2.75e-52	1.75e-49	1.44e-49				8
Ube2c~	43								9
## 3 GO:000~ sister chr~	39/127	181/23~	2.95e-52	1.75e-49	1.44e-49				10
Ube2c~	39								11
## 4 GO:014~ mitotic nu~	43/127	268/23~	7.78e-52	2.91e-49	2.39e-49				12
Ube2c~	43								13
## 5 GO:000~ mitotic si~	37/127	151/23~	8.18e-52	2.91e-49	2.39e-49				14
Ube2c~	37								15
## 6 GO:000~ nuclear di~	46/127	418/23~	9.43e-48	2.80e-45	2.30e-45				16
Ube2c~	46								17
## 7 GO:004~ organelle ~	47/127	472/23~	9.12e-47	2.32e-44	1.91e-44				18
Mtfr2~	47								19
## 8 GO:000~ spindle or~	27/127	179/23~	1.59e-31	3.54e-29	2.91e-29				20
Aurka~	27								21
## 9 GO:190~ microtubul~	24/127	142/23~	2.56e-29	5.06e-27	4.16e-27				22
Aurka~	24								23
## 10 GO:005~ regulation~	21/127	103/23~	1.42e-27	2.53e-25	2.08e-25				24
Ube2c~	21								25
## # ... with 1,770 more rows									26

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

```

	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
	geneID	Count	<chr>	<chr>	<dbl>	<dbl>	<dbl>
	chr	chr	<int>				
## 1	mmu04210	Apoptosis	15/141	136/89~	2.94e-9	6.71e-7	4.65e-7
	Ctsc	/~	15				
## 2	mmu04145	Phagosome	15/141	182/89~	1.55e-7	1.76e-5	1.22e-5
	Tubb5	/~	15				
## 3	mmu05166	Human T-cel~	17/141	250/89~	3.58e-7	2.39e-5	1.65e-5
	Il1r2	/~	17				
## 4	mmu05202	Transcripti~	16/141	224/89~	4.19e-7	2.39e-5	1.65e-5
	Il1r2	/~	16				
## 5	mmu04640	Hematopoiet~	10/141	94/8949	2.00e-6	9.14e-5	6.33e-5
	Il1r2	/~	10				
## 6	mmu04380	Osteoclast ~	11/141	128/89~	5.07e-6	1.93e-4	1.33e-4
	Fosl2	/~	11				
## 7	mmu05323	Rheumatoid ~	9/141	87/8949	8.38e-6	2.73e-4	1.89e-4
	Ctsk	/~	9				
## 8	mmu05140	Leishmanias~	8/141	70/8949	1.29e-5	3.68e-4	2.55e-4
	Itga4	/~	8				
## 9	mmu05152	Tuberculosis	12/141	180/89~	2.53e-5	6.42e-4	4.45e-4
	Lsp1	/~	12				
## 10	mmu04064	NF-kappa B ~	9/141	105/89~	3.86e-5	8.81e-4	6.10e-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

```

	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
	geneID	Count	<chr>	<chr>	<dbl>	<dbl>	<dbl>
	chr	chr	<int>				
## 1	GO:0050~	regulation~	23/224	372/23~	1.72e-12	5.89e-9	4.30e-9
	Il1r2	/~	23				

## 2 GO:0002~ negative r~ 22/224	462/23~	8.04e-10	7.67e-7	5.61e-7	Fgr	5
/C~ 22						
## 3 GO:0022~ positive r~ 17/224	265/23~	8.72e-10	7.67e-7	5.61e-7		6
Ceaca~ 17						
## 4 GO:1903~ regulation~ 21/224	424/23~	9.85e-10	7.67e-7	5.61e-7		7
Ceaca~ 21						
## 5 GO:0007~ leukocyte ~ 19/224	345/23~	1.12e- 9	7.67e-7	5.61e-7		8
Ceaca~ 19						
## 6 GO:0050~ leukocyte ~ 19/224	360/23~	2.27e- 9	1.29e-6	9.44e-7		9
Itga4~ 19						
## 7 GO:0050~ regulation~ 19/224	372/23~	3.88e- 9	1.75e-6	1.28e-6		10
Ceaca~ 19						
## 8 GO:1903~ positive r~ 15/224	221/23~	4.10e- 9	1.75e-6	1.28e-6		11
Ceaca~ 15						
## 9 GO:0072~ lymphocyte~ 11/224	103/23~	4.60e- 9	1.75e-6	1.28e-6		12
Itga4~ 11						
## 10 GO:0019~ antigen pr~ 6/224	16/233~	5.42e- 9	1.85e-6	1.35e-6		13
Ctss/~ 6						
## # ... with 3,409 more rows						14

6.2.7 GO enrichment for MHCII IM specific genes + common genes class 3

As the specific genes have limited number. We try add the class 3 genes, which are also terminated differentiation genes. But we have to pay attention that the class 3 gene number is huge compare to the specific genes, so enrichment results will be diluted.

```
symb <- c(geneSymbol.changed.mhcii, genes.class3)
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
  , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
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_
  MHCIIISpecific_plus_Class3_common.csv")
result.enrichGO@result
```

## # A tibble: 3,607 x 9	1						
ID Description GeneRatio BgRatio pvalue p.adjust qvalue	2						
geneID Count	3						
<chr> <chr> <chr> <dbl> <dbl> <dbl> <	3						
chr> <int>							
## 1 GO:005~ regulation~ 28/260	372/23~	1.81e-15	4.10e-12	2.84e-12			4
Cd24a~ 28							
## 2 GO:000~ leukocyte ~ 27/260	345/23~	2.28e-15	4.10e-12	2.84e-12			5
Itgb7~ 27							
## 3 GO:002~ positive r~ 24/260	265/23~	3.41e-15	4.10e-12	2.84e-12			6
Cd24a~ 24							
## 4 GO:190~ positive r~ 22/260	221/23~	7.08e-15	6.38e-12	4.41e-12			7
Cd24a~ 22							
## 5 GO:001~ antigen pr~ 9/260	16/233~	2.47e-14	1.78e-11	1.23e-11	H2-		8
DM~ 9							
## 6 GO:005~ leukocyte ~ 26/260	360/23~	5.13e-14	3.08e-11	2.13e-11			9
Itgb7~ 26							
## 7 GO:000~ positive r~ 28/260	449/23~	1.99e-13	1.03e-10	7.09e-11			10
Wnt11~ 28							

```

## 8 GO:005~ regulation~ 24/260      321/23~ 2.38e-13 1.07e-10 7.40e-11 11
Cd24a~ 24
## 9 GO:190~ regulation~ 27/260      424/23~ 3.35e-13 1.20e-10 8.30e-11 12
Cd24a~ 27
## 10 GO:005~ positive r~ 29/260     496/23~ 3.65e-13 1.20e-10 8.30e-11 13
Cd24a~ 29
## # ... with 3,597 more rows          14

```

6.2.8 GO enrichment for CD206 IM specific genes + common genes class 3

```

symb <- c(geneSymbol.changed.cd206, genes.class3)
de_entrez <- bitr( geneID = symb, fromType = "SYMBOL", toType = "ENTREZID"
  , OrgDb = "org.Mm.eg.db", drop = TRUE ) $ ENTREZID
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_
  CD206Specific_plus_Class3_common.csv")
result.enrichGO@result

```

	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
	geneID	Count	<chr>	<chr>	<dbl>	<dbl>	<dbl>
	chr	chr	<chr>	<chr>	<dbl>	<dbl>	<dbl>
##	1	GO:005~ leukocyte	~ 36/294	360/23~	6.24e-22	2.38e-18	1.60e-18
	Trem2~	36					
##	2	GO:009~ myeloid le~	25/294	219/23~	6.62e-17	1.26e-13	8.50e-14
	Trem2~	25					
##	3	GO:006~ cell chemo~	28/294	303/23~	2.25e-16	2.86e-13	1.93e-13
	Cxcl2~	28					
##	4	GO:003~ leukocyte	~ 24/294	219/23~	7.21e-16	6.87e-13	4.63e-13
	Cxcl2~	24					
##	5	GO:007~ lymphocyte~	17/294	103/23~	1.39e-14	1.06e-11	7.16e-12
	Ecm1/~	17					
##	6	GO:005~ regulation~	28/294	372/23~	4.15e-14	2.26e-11	1.52e-11
	Ctsl/~	28					
##	7	GO:005~ regulation~	28/294	372/23~	4.15e-14	2.26e-11	1.52e-11
	Trem2~	28					
##	8	GO:000~ regulation~	22/294	217/23~	6.10e-14	2.91e-11	1.96e-11
	Trem2~	22					
##	9	GO:007~ ERK1 and E~	26/294	325/23~	8.50e-14	3.60e-11	2.42e-11
	Trem2~	26					
##	10	GO:003~ response t~	18/294	137/23~	1.40e-13	5.32e-11	3.58e-11
	Irf8/~	18					
## # ... with 3,800 more rows							

7 Show gene expression pattern with TradeSeq results

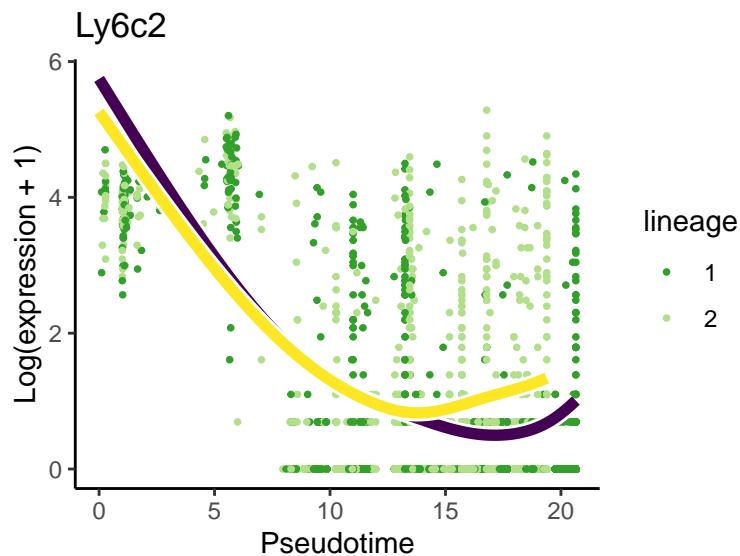
7.1 Class 1 common genes

```

require(ggplot2)
sigGene <- "Ly6c2"

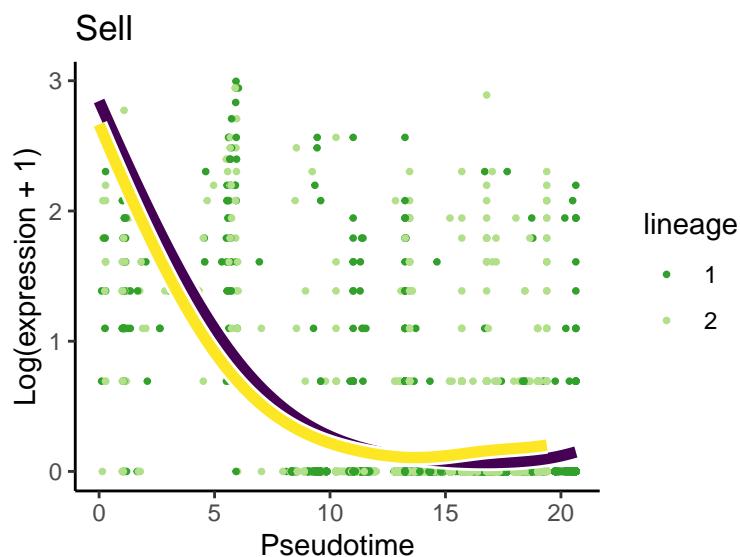
```

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

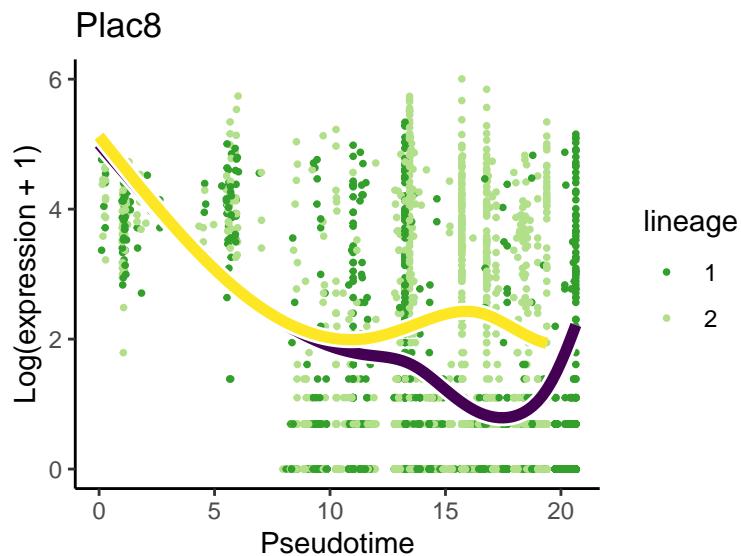


```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           ".pdf"),
#         width = 4, height = 3)
```

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



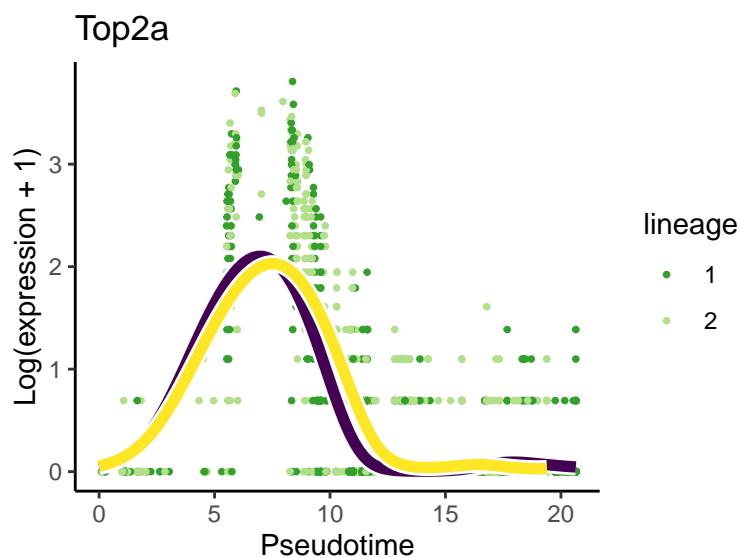
```
sigGene <- "Plac8"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```



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

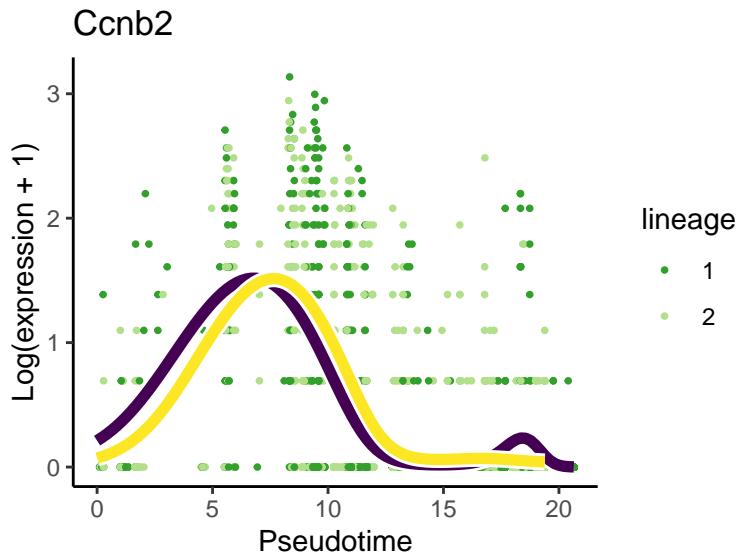
7.2 Class 2 common genes

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



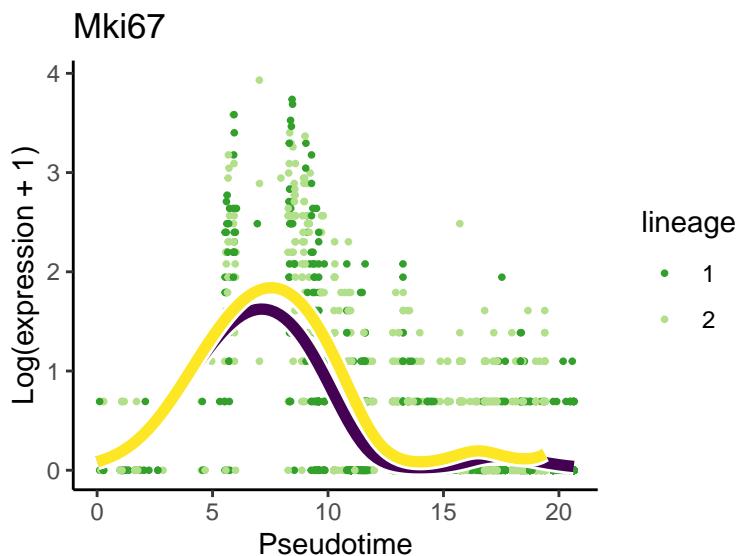
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3)
```

```
sigGene <- "Ccnb2"
plotSmothers(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) 1
#                           2
```

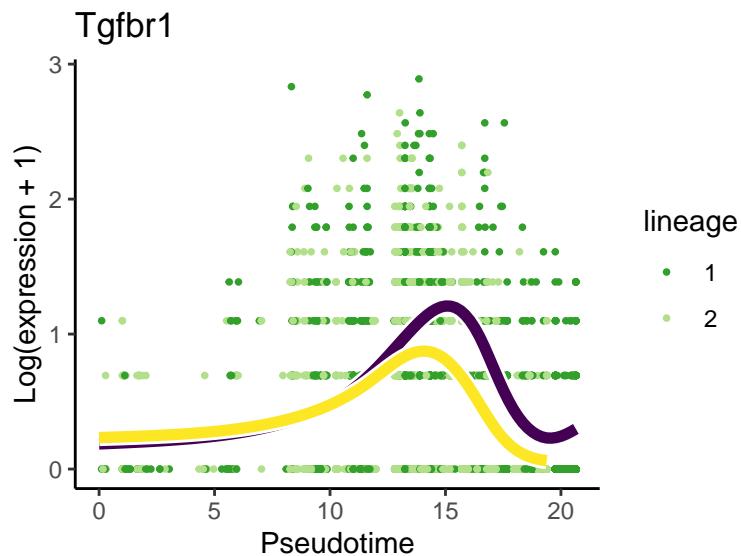
```
sigGene <- "Mki67"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 1
  2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           2
```

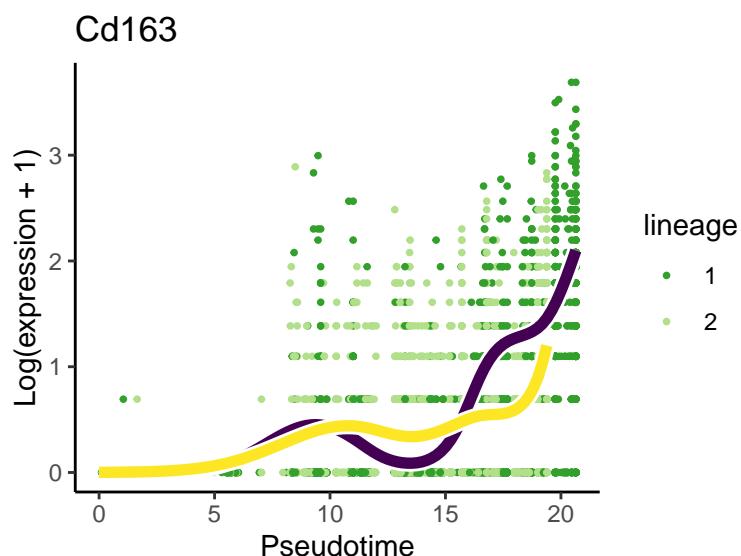
7.3 Class 3 common genes

```
sigGene <- "Tgfbri1"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 1
  2
```



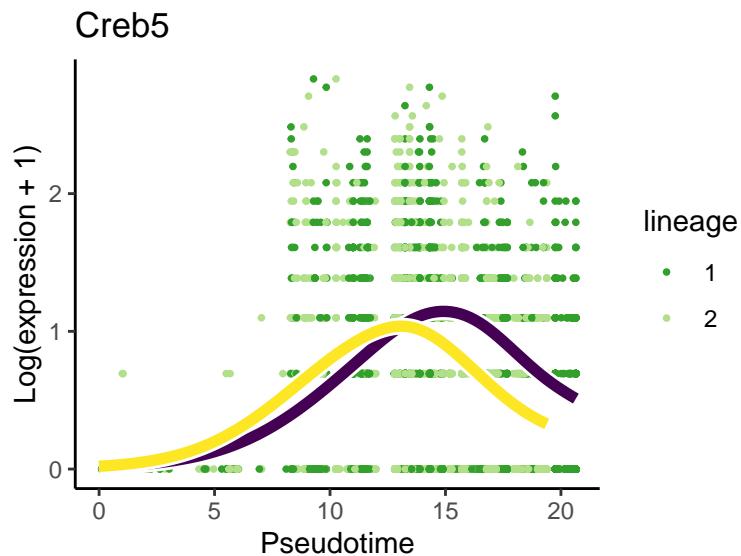
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           ".pdf"),
#         width = 4, height = 3) 1
# 2
```

```
sigGene <- "Cd163"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



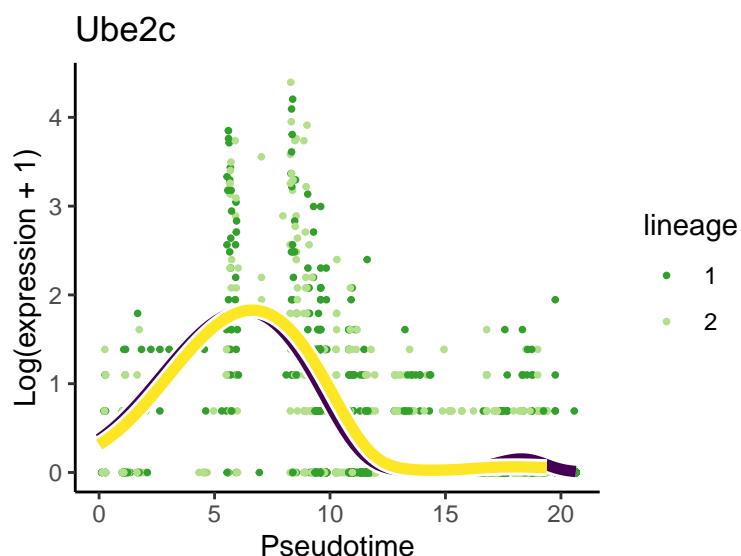
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           ".pdf"),
#         width = 4, height = 3) 1
# 2
```

```
sigGene <- "Creb5"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                   ".pdf"),
#       width = 4, height = 3) 1
# 2
```

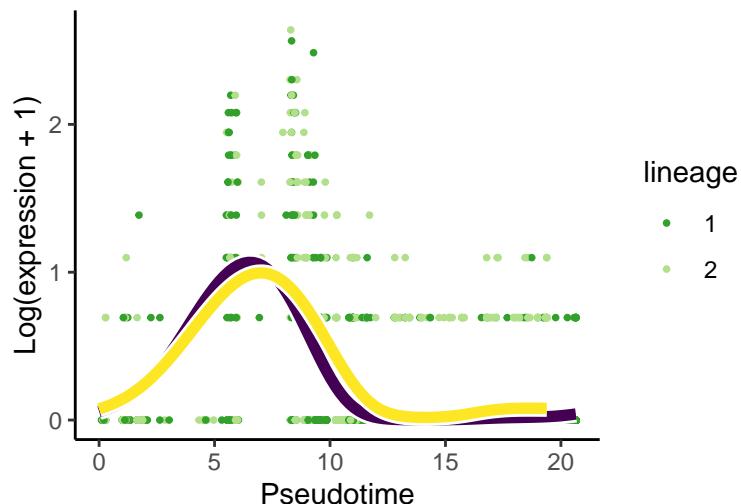
```
sigGene <- "Ube2c"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene) 1
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                   ".pdf"),
#       width = 4, height = 3) 1
# 2
```

```
sigGene <- "Aurkb"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene) 1
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 2
```

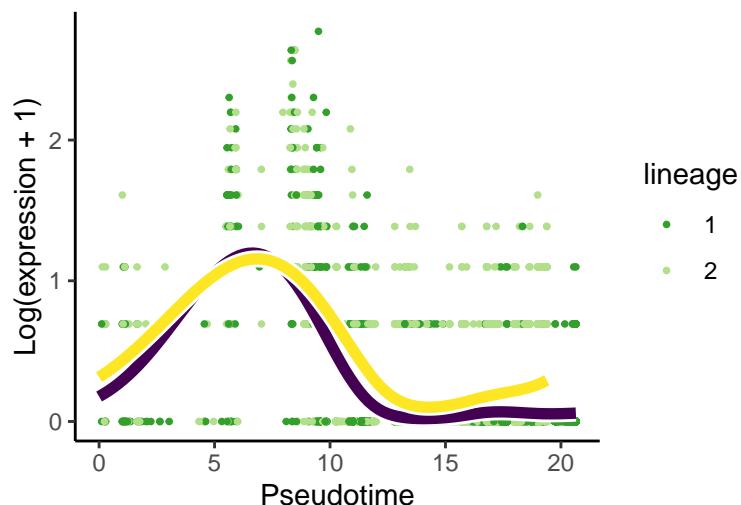
Aurkb



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

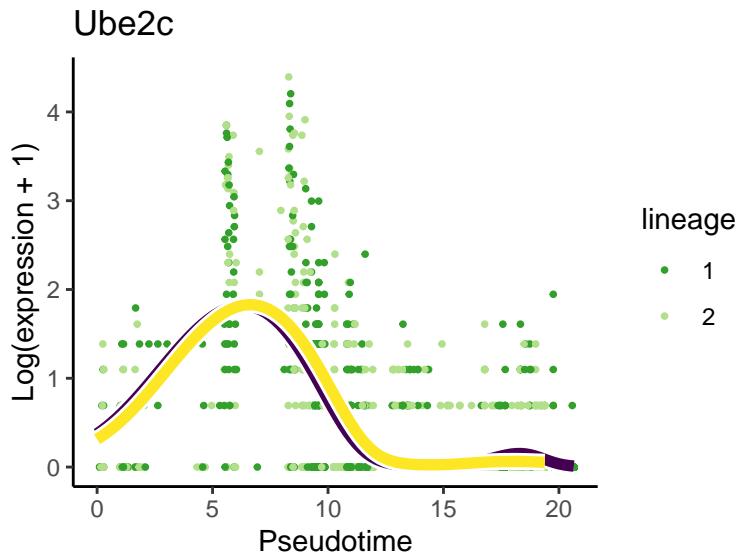
```
sigGene <- "Racgap1"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```

Racgap1



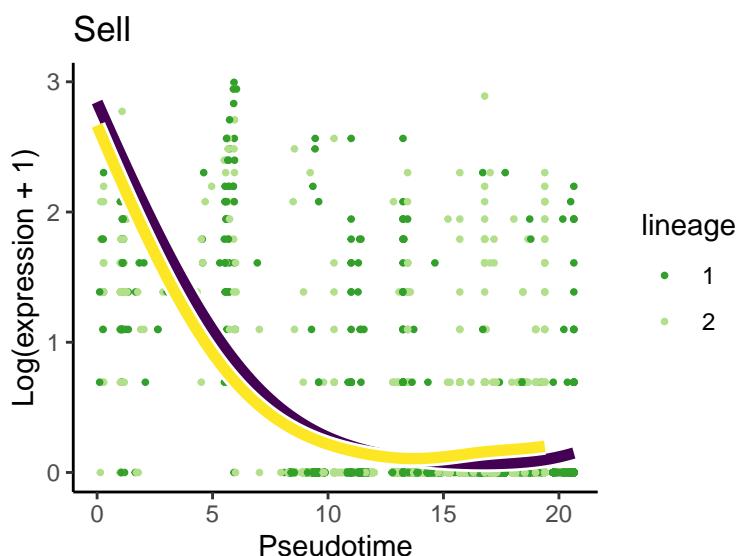
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3)
```

```
sigGene <- "Ube2c"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```



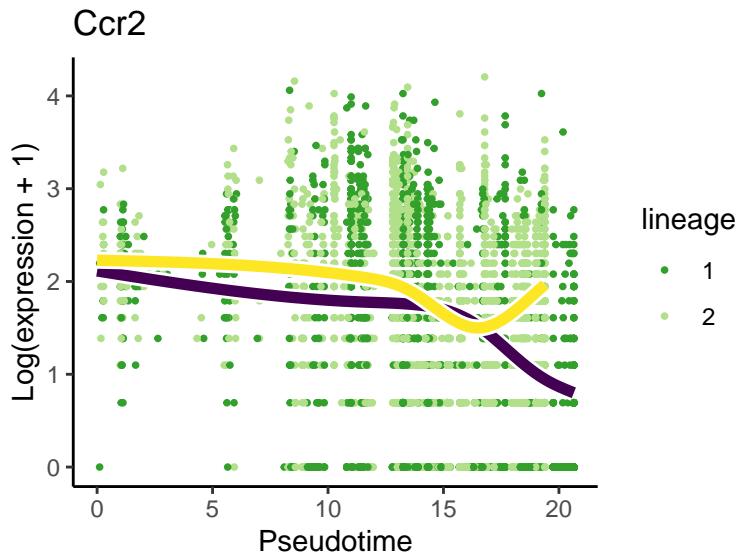
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           2
```

```
sigGene <- "Sell"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 1
  2
```

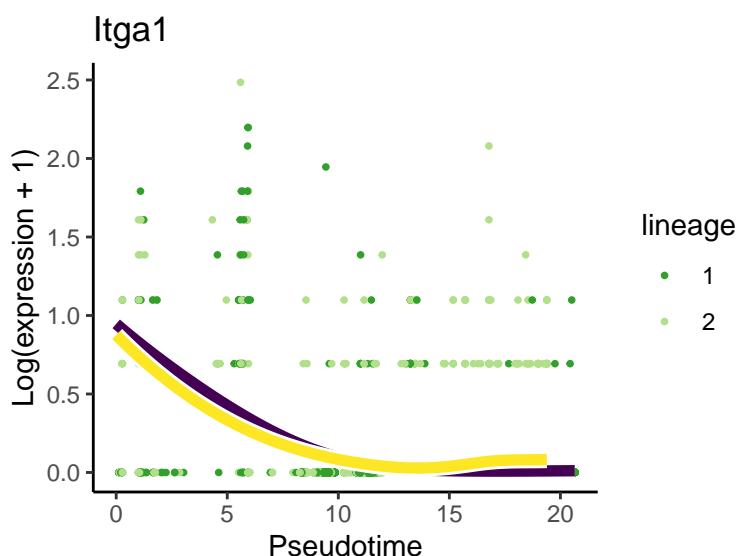


```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           2
```

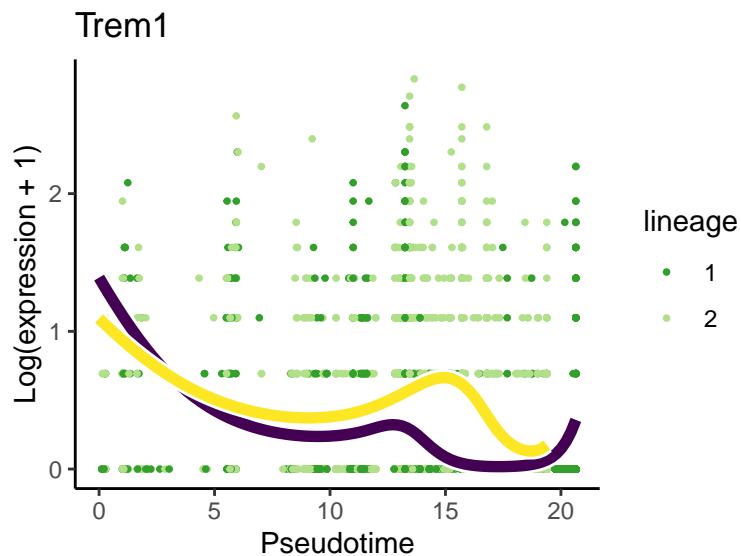
```
sigGene <- "Ccr2"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values = c("#33A02C", "#B2DF8A")) 1
  2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           "sigGene",
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           "Itga1"),
# plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 2
# ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```

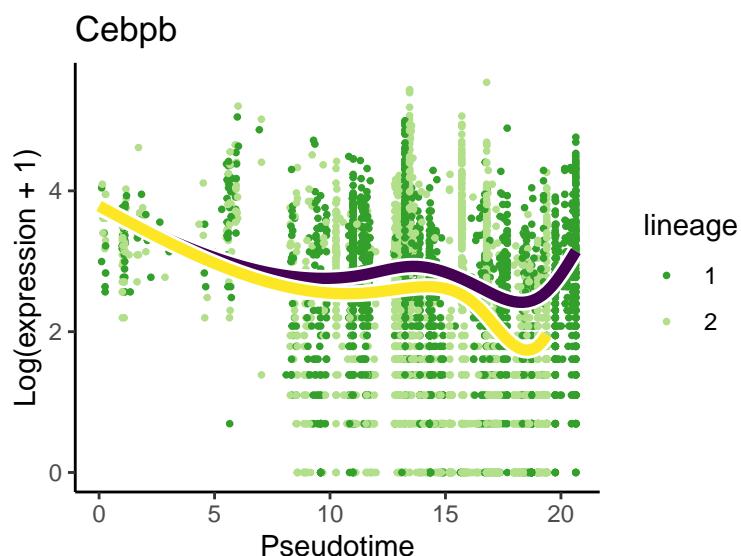


```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           "sigGene",
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           "Trem1"),
# plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 2
# ) + scale_color_manual(values =c("#33A02C", "#B2DF8A"))
```



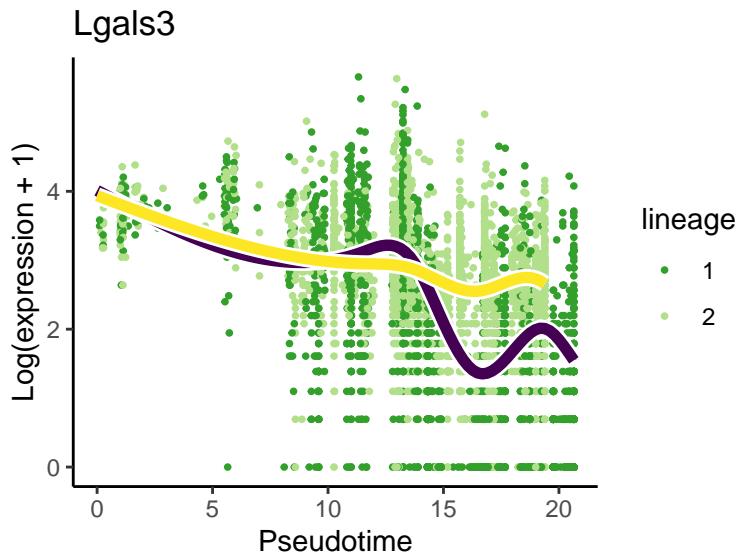
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           ".pdf"),
#         width = 4, height = 3) 1
# 2
```

```
sigGene <- "Cebpb"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



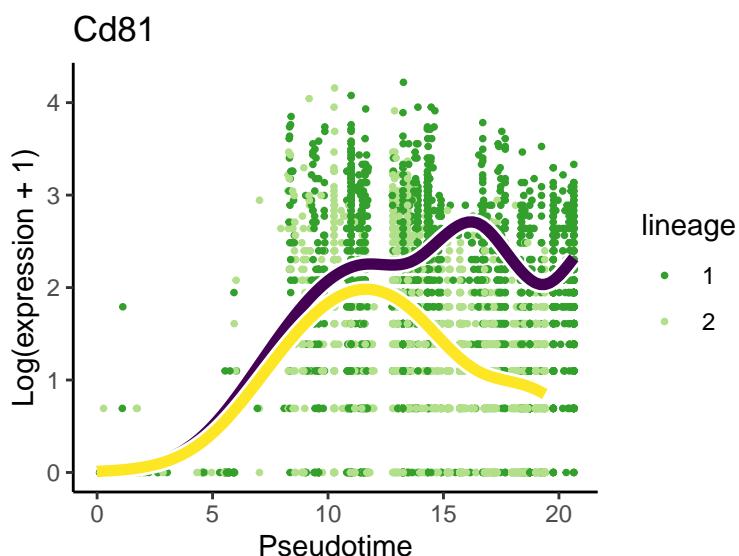
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           ".pdf"),
#         width = 4, height = 3) 1
# 2
```

```
sigGene <- "Lgals3"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



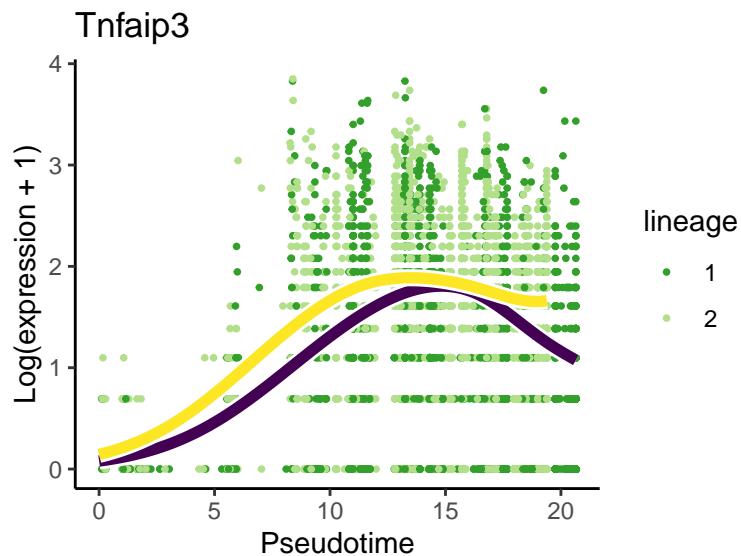
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           2
```

```
sigGene <- "Cd81"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 1
  2
```



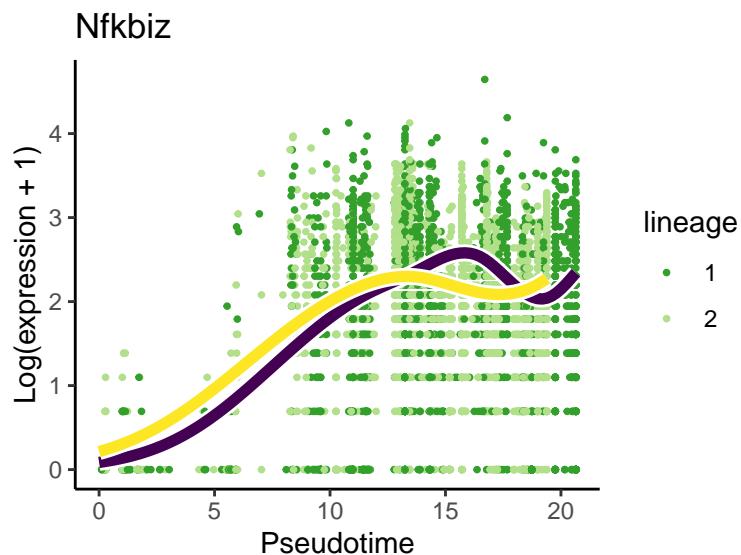
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
#                           sigGene,
#                           ".pdf"),
#         width = 4, height = 3) 1
#                           2
```

```
sigGene <- "Tnfaip3"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene)
  + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 1
  2
```



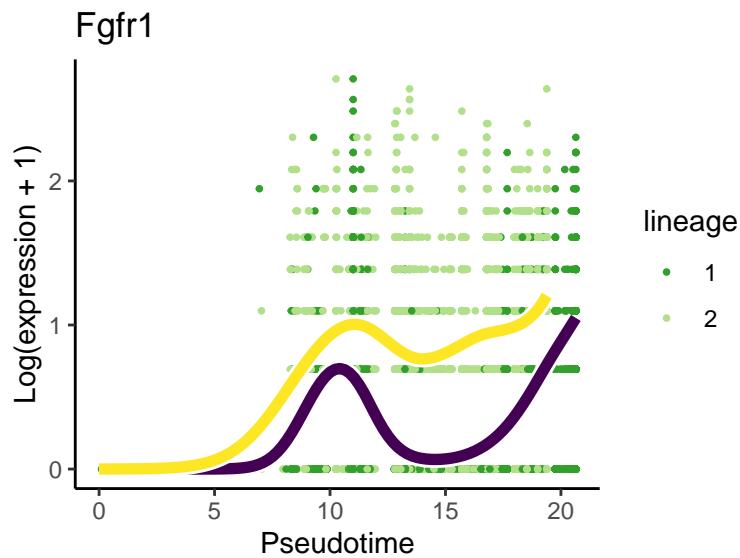
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Nfkbiz"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



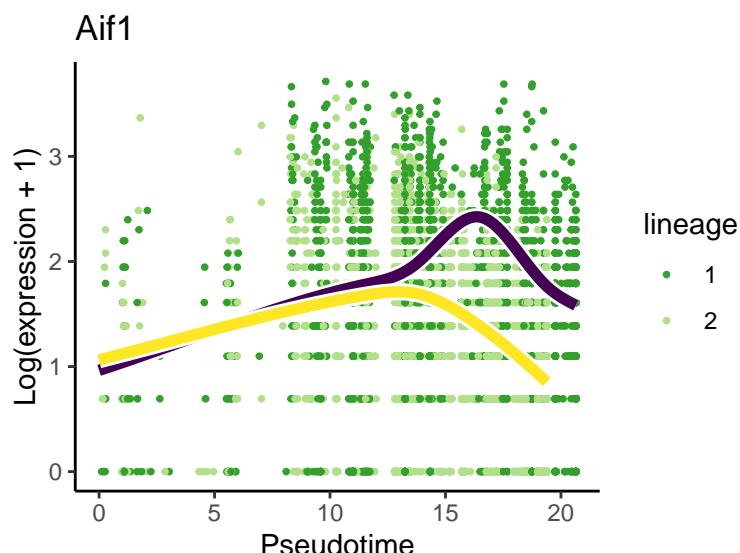
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Fgfr1"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



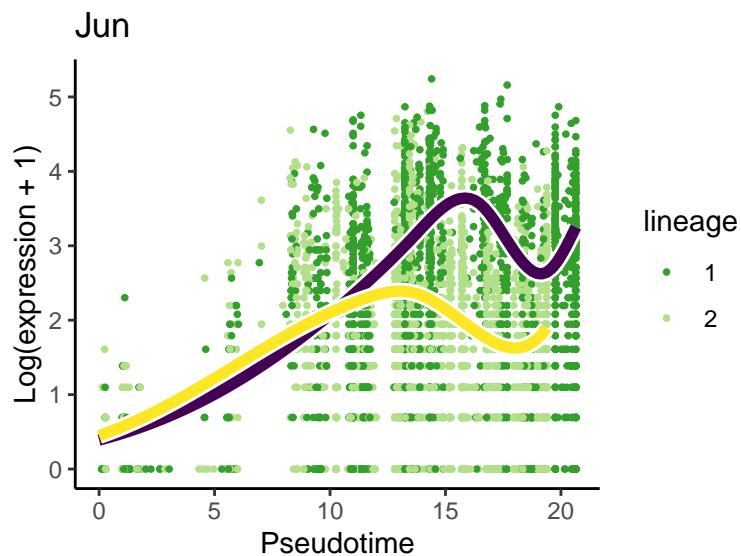
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

```
sigGene <- "Aif1"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



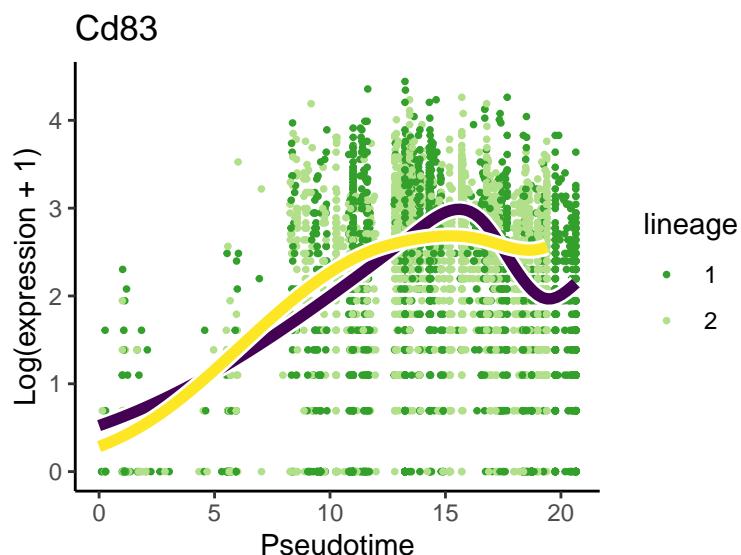
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

```
sigGene <- "Jun"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



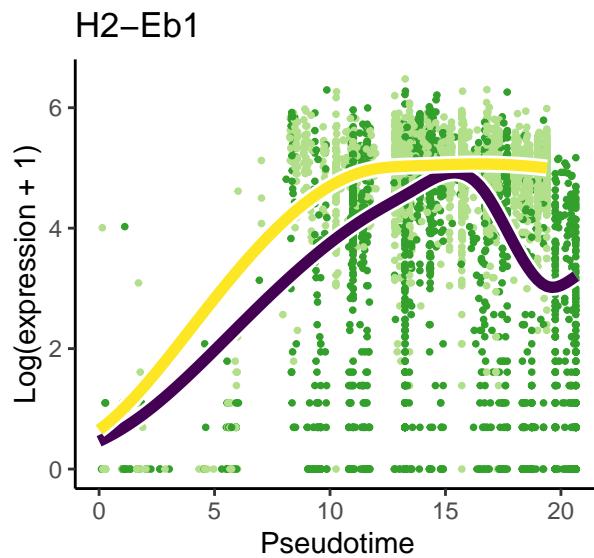
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Cd83"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



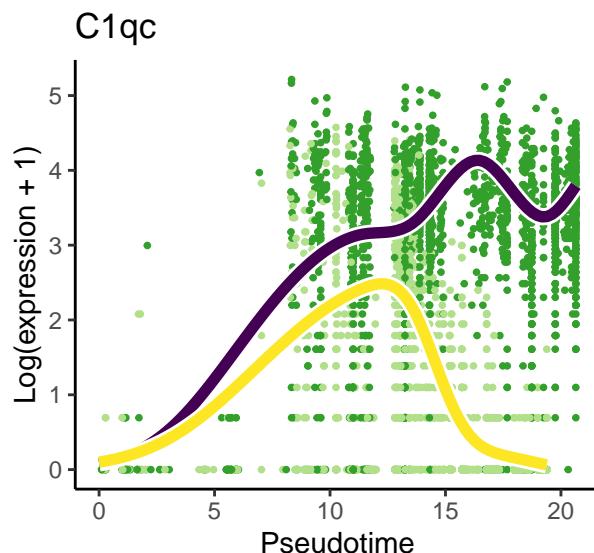
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "H2-Eb1"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



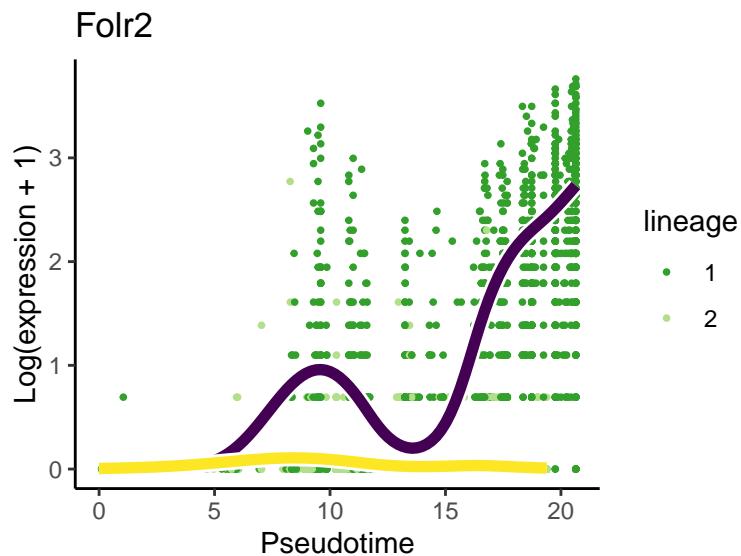
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

```
sigGene <- "C1qc"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



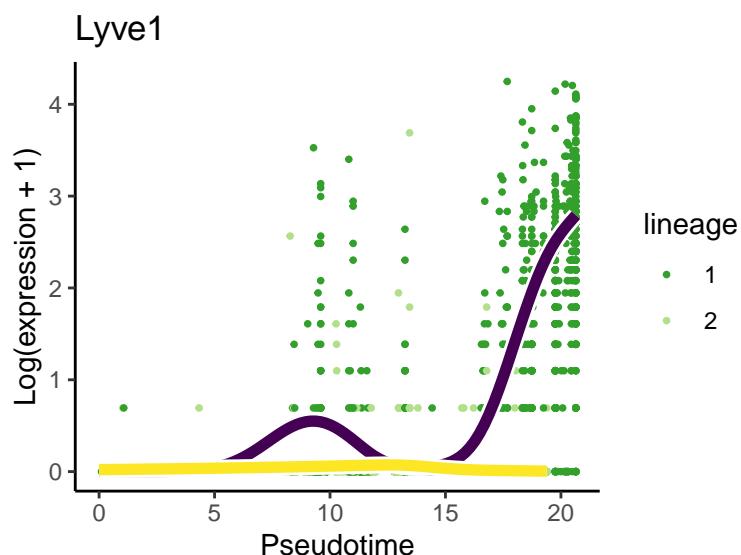
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

```
sigGene <- "Folr2"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



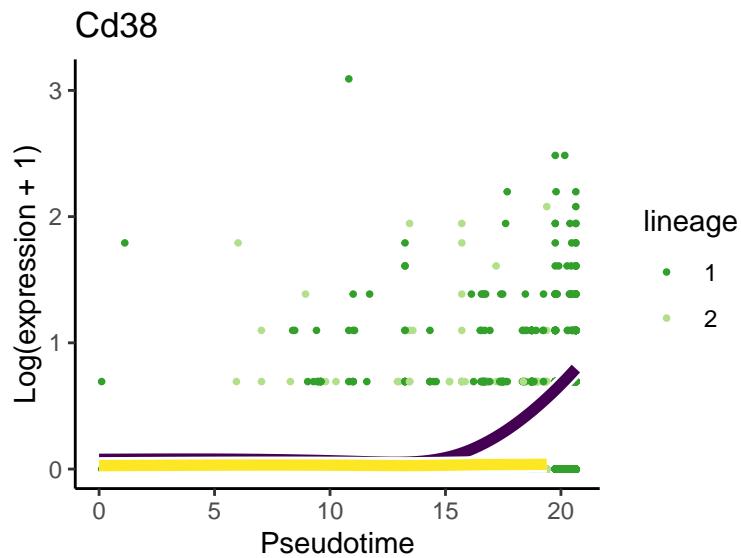
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Lyve1"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



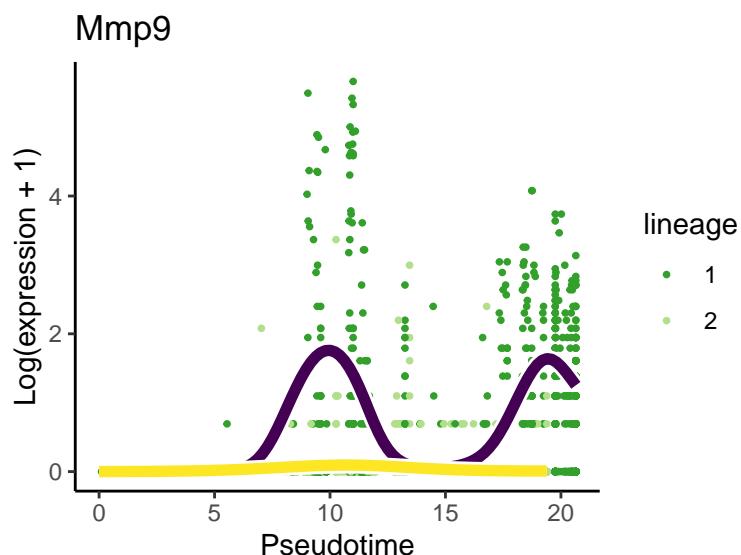
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Cd38"
plotSmoothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



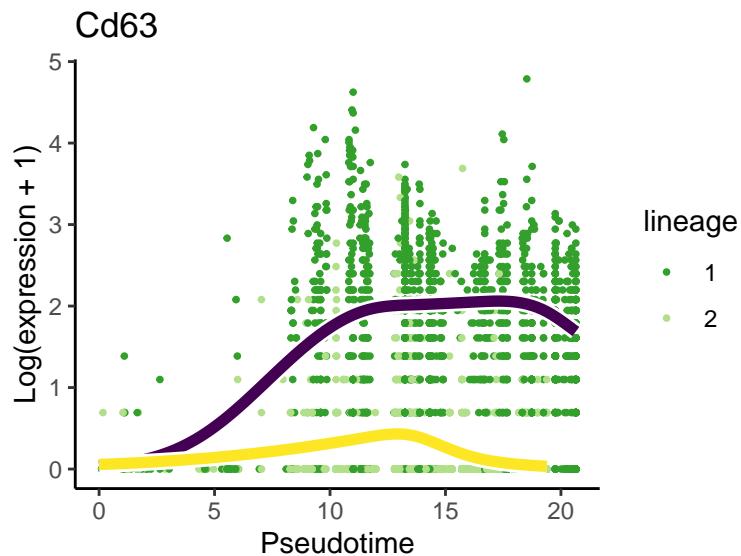
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Mmp9"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



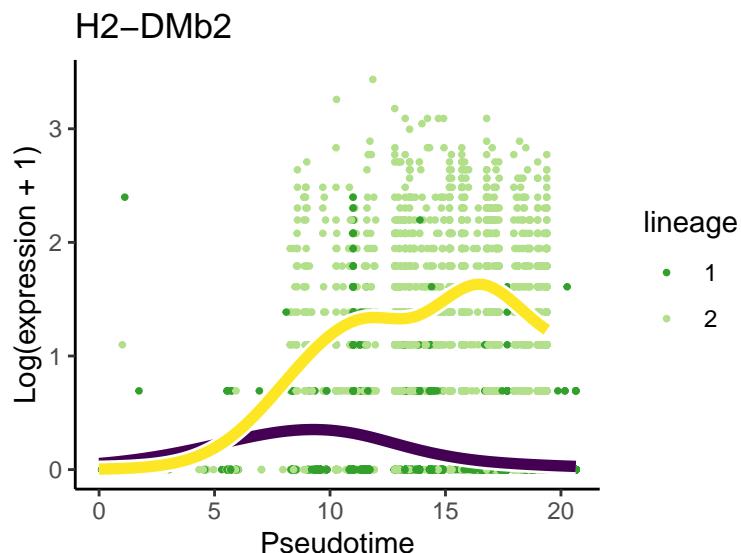
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Cd63"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

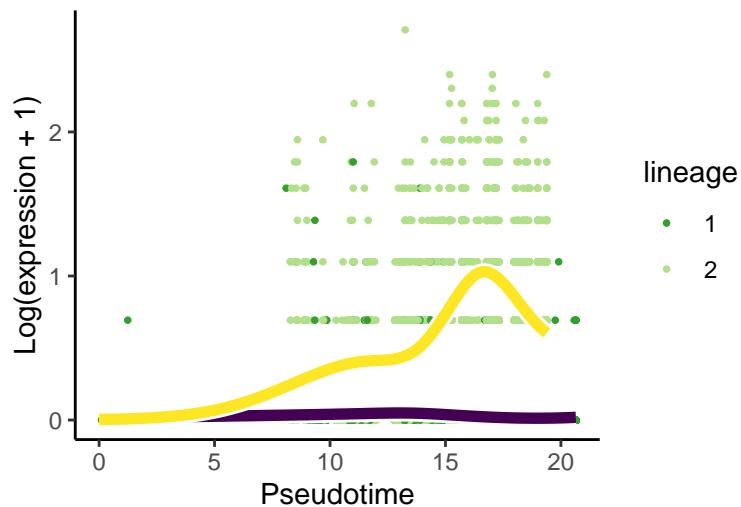
```
sigGene <- "H2-DMb2"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene,
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "H2-0a"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```

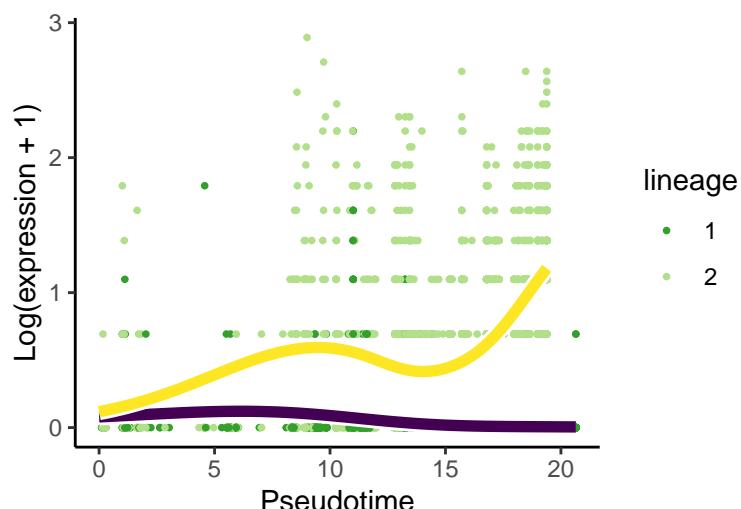
H2-Oa



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

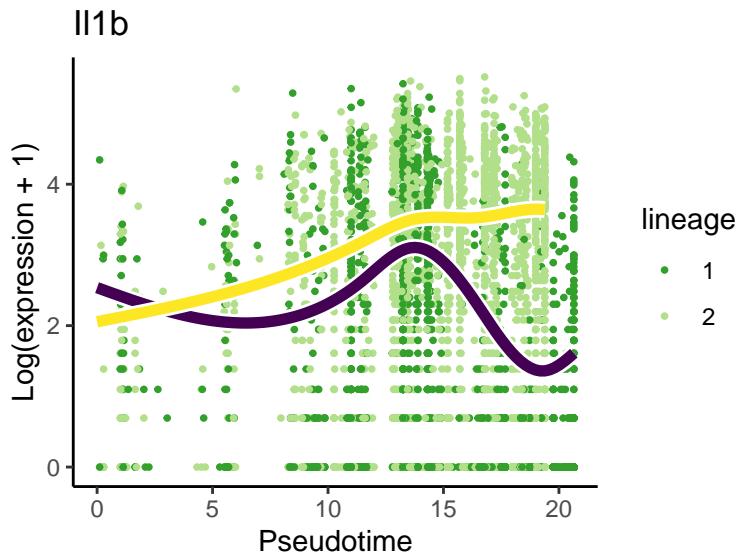
```
sigGene <- "Wnt11"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```

Wnt11



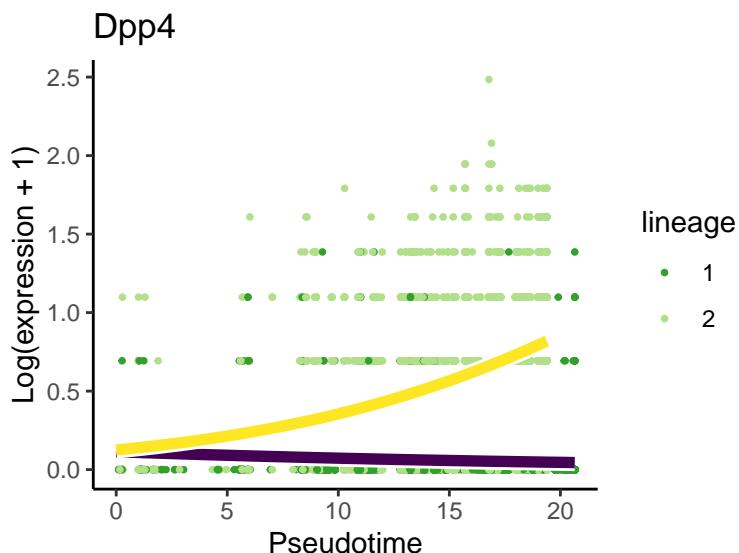
```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene
, ".pdf"), width = 4, height = 3) 1
```

```
sigGene <- "Il1b"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

```
sigGene <- "Dpp4"
plotSmothers(sce, counts = counts(sce), gene = sigGene) + ggtitle(sigGene 1
) + scale_color_manual(values =c("#33A02C", "#B2DF8A")) 2
```



```
# ggsave(filename = paste("../Figures/TradeSeq_plot_IM_lineages_",
, sigGene 1
, ".pdf"), width = 4, height = 3)
```

8 Session information

R session:

```
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_2.0.2
## [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_0.4.14
## [19] RColorBrewer_1.1-2            dplyr_1.0.8
## [21] ggplot2_3.3.5                ComplexHeatmap_2.6.2
## [23] SeuratObject_4.0.4            Seurat_4.1.0
##
## loaded via a namespace (and not attached):
## [1] scattermore_0.8              princurve_2.1.6      pkgmaker_0.32.2
## [4] tidyrr_1.2.0                 bit64_4.0.5          knitr_1.37
## [7] irlba_2.3.5                 DelayedArray_0.16.3  data.table_1.14.2
## [10] rpart_4.1-15                 RCurl_1.98-1.6       doParallel_1.0.17
## [13] generics_0.1.2               terra_1.5-21         cowplot_1.1.1
## [16] RSQLite_2.2.10               shadowtext_0.1.1     RANN_2.6.1
## [19] VGAM_1.1-6                  combinat_0.0-8       proxy_0.4-26
## [22] future_1.24.0               enrichplot_1.10.2   bit_4.0.4
## [25] phylobase_0.8.10            spatstat.data_2.1-2 xml2_1.3.3
## [28] httpuv_1.6.5                wk_0.6.0             assertthat_0.2.1
## [31] viridis_0.6.2                xfun_0.29            hms_1.1.1
## [34] evaluate_0.15                promises_1.2.0.1    fansi_1.0.2
## [37] progress_1.2.2               igraph_1.2.11        DBI_1.1.2
## [40] htmlwidgets_1.5.4            sparsesvd_0.2        spatstat.geom_2.3-2
## [43] spdep_1.2-2                 purrr_0.3.4          ellipsis_0.3.2
## [46] DDRTree_0.1.5                annotate_1.68.0     gridBase_0.4-7
## [49] locfdr_1.1-8                 deldir_1.0-6         vctrs_0.3.8
## [52] Cairo_1.5-14                ROCR_1.0-11         abind_1.4-5

```

## [55]	cachem_1.0.6	withr_2.4.3	ggforce_0.3.3	54
## [58]	sctransform_0.3.3	prettyunits_1.1.1	goftest_1.2-3	55
## [61]	DOSE_3.16.0	softImpute_1.4-1	cluster_2.1.0	56
## [64]	ape_5.6-1	lazyeval_0.2.2	crayon_1.5.0	57
## [67]	genefilter_1.72.1	edgeR_3.32.1	pkgconfig_2.0.3	58
## [70]	slam_0.1-50	labeling_0.4.2	units_0.8-0	59
## [73]	tweenr_1.0.2	nlme_3.1-155	rlang_1.0.1	60
## [76]	globals_0.14.0	lifecycle_1.0.1	miniUI_0.1.1.1	61
## [79]	downloader_0.4	registry_0.5-1	rsvd_1.0.5	62
## [82]	polyclip_1.10-0	lmtest_0.9-39	rngtools_1.5.2	63
## [85]	Matrix_1.4-0	raster_3.5-15	Rhdf5lib_1.12.1	64
## [88]	boot_1.3-25	zoo_1.8-9	ggridges_0.5.3	65
## [91]	GlobalOptions_0.1.2	pheatmap_1.0.12	png_0.1-7	66
## [94]	viridisLite_0.4.0	rjson_0.2.21	bitops_1.0-7	67
## [97]	rhdf5filters_1.2.1	rncl_0.8.4	KernSmooth_2.23-20	68
## [100]	blob_1.2.2	shape_1.4.6	classInt_0.4-3	69
## [103]	qvalue_2.22.0	stringr_1.4.0	slingshot_1.8.0	70
## [106]	zinbwave_1.12.0	s2_1.0.7	parallelly_1.30.0	71
## [109]	spatstat.random_2.1-0	beachmat_2.6.4	scales_1.1.1	72
## [112]	memoise_2.0.1	plyr_1.8.6	ica_1.0-2	73
## [115]	howmany_0.3-1	zlibbioc_1.36.0	scatterpie_0.1.7	74
## [118]	compiler_4.0.3	HSMSingleCell_1.10.0	clue_0.3-60	75
## [121]	fitdistrplus_1.1-6	cli_3.2.0	ade4_1.7-18	76
## [124]	XVector_0.30.0	listenv_0.8.0	patchwork_1.1.1	77
## [127]	pbapply_1.5-0	MASS_7.3-53	mgcv_1.8-33	78
## [130]	tidyselect_1.1.1	stringi_1.7.6	highr_0.9	79
## [133]	densityClust_0.3.1	GOSemSim_2.16.1	yaml_2.3.5	80
## [136]	BiocSingular_1.6.0	locfit_1.5-9.4	ggrepel_0.9.1	81
## [139]	pbmccapply_1.5.0	fastmatch_1.1-3	tools_4.0.3	82
## [142]	future.apply_1.8.1	rstudioapi_0.13	uuid_1.0-3	83
## [145]	monocle_2.18.0	foreach_1.5.2	RNeXML_2.4.6	84
## [148]	gridExtra_2.3	farver_2.1.0	Rtsne_0.15	85
## [151]	ggraph_2.0.5	BiocManager_1.30.16	rvcheck_0.2.1	86
## [154]	digest_0.6.29	FNN_1.1.3	shiny_1.7.1	87
## [157]	qlcMatrix_0.9.7	Rcpp_1.0.8	later_1.3.0	88
## [160]	RcppAnnoy_0.0.19	httr_1.4.2	sf_1.0-6	89
## [163]	kernlab_0.9-29	colorspace_2.0-3	XML_3.99-0.8	90
## [166]	tensor_1.5	reticulate_1.24	splines_4.0.3	91
## [169]	yulab.utils_0.0.4	uwot_0.1.11	spatstat.utils_2	92
.3-0				
## [172]	graphlayouts_0.8.0	sp_1.4-6	plotly_4.10.0	93
## [175]	spData_2.0.1	xtable_1.8-4	jsonlite_1.7.3	94
## [178]	tidygraph_1.2.0	ggfun_0.0.5	R6_2.5.1	95
## [181]	pillar_1.7.0	htmltools_0.5.2	mime_0.12	96
## [184]	NMF_0.23.0	glue_1.6.1	fastmap_1.1.0	97
## [187]	BiocParallel_1.24.1	class_7.3-17	codetools_0.2-18	98
## [190]	fgsea_1.16.0	utf8_1.2.2	lattice_0.20-41	99
## [193]	spatstat.sparse_2.1-0	tibble_3.1.6	leiden_0.3.9	100
## [196]	magick_2.7.3	GO.db_3.12.1	survival_3.2-7	101
## [199]	limma_3.46.0	rmarkdown_2.11	doctopt_0.7.1	102
## [202]	fastICA_1.2-3	munsell_0.5.0	D0.db_2.9	103
## [205]	rhdf5_2.34.0	e1071_1.7-9	GetoptLong_1.0.5	104
## [208]	GenomeInfoDbData_1.2.4	iterators_1.0.14	HDF5Array_1.18.1	105
## [211]	reshape2_1.4.4	gtable_0.3.0	spatstat.core_2.4-0	106

References

1. Berge, K. V. den *et al.* Trajectory-based differential expression analysis for single-cell sequencing data. *Nature Communications* 2020 **11**:1, 1–13 (2020).
2. Gu, Z., Eils, R. & Schlesner, M. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics (Oxford, England)* **32**, 2847–2849 (2016).