

Pseudotime for clusters 0 and 2

Ramiro Lorenzo

April 8, 2020

Load library

```
library(monocle3,verbose = FALSE)
library(dplyr,verbose = FALSE)
```

Read the data

```
## Loads a sparse matrix RawCountsPseudotime
load("../RawCountsPseudotime.rda")
dim(RawCountsPseudotime)

## [1] 20271 3127
```

Load cluster data

```
## Loads a factor variable ClusterPseudotime containing cluster identities
load("../ClusterPseudotime.rda")
## Change the name of variable to remind
DataClusters <- ClusterPseudotime

table(DataClusters)
```

```
## DataClusters
##      0.0      0.1      1.0      1.1     10.0     10.1     10.2
##    238      213      410      30       0       0      34
## 11.0_1_2 11.3 11.4 12.0 12.1 13.0 14.0
##      0       0       0       0       0       0       0
##     16.0     16.1     17.0     17.1     19.0     19.1     2.0
##     160      42      169      36       0       0     383
##    20.0     22.0     22.1 22.2_3     22.4     26.0     26.1
##      0       0       0       0       0       0       0
##   26.2 27.0_3 27.1 27.2 27.NA 28.0_2_3_4     28.1
##      0       0       0       0       0       0       0
##     29.0     29.1     3.0      3.1     30.0     31.0     33.0
##      0       0     253     119       0       0      90
##    34.0     35.0     35.1     36.0     37.0     37.1     38.0
##      0       0       0       0       0       0       0
##    39.0      4.0      4.1     40.0     41.0     42.0     42.1
##      0     206     142       0       0      40       27
```

```

##      43.0     44.0     45.0     46.0     48.0     49.0     5.0
##      0         0         0         0         0         0         0
##      5.1     50.0     51.0     51.1     52.0     53.0     54.0
##      0         0         0         0         0         0         0
##      55.0     56.0     57.0     59.0      6.0     60.0     61.0
##      0         43        0         0         0         0         0
##      62.0     63.0      7.0      8.0      8.1      8.2      8.3
##      0         0         0         0        77        27       15
##  10.0.0  10.0.1_4  10.0.2_3  10.0.5  10.1.0  10.1.1  8.0.0
##      45        59        57        9        33        21      103
##      8.0.1
##      46

length(DataClusters)

```

```
## [1] 3127
```

Convert DataClusters to a matrix format for input to Monocle

```

DataCluster.ID <- matrix(as.character(levels(DataClusters))[DataClusters], ncol = 1)
rownames(DataCluster.ID) <- names(DataClusters)
colnames(DataCluster.ID) <- "Cluster.IDs"
DataCluster.ID[1:10,]

## cele-001-008.GATCAGTCAT cele-001-027.ACTCCGCCAA cele-001-042.TTCCTAGACC
##          "0.0"           "0.0"           "0.0"
## cele-001-046.TTCTACGCCA cele-001-047.TTCGCTGCCT cele-001-047.ATGGAAGCAT
##          "0.0"           "0.0"           "0.0"
## cele-001-064.AAGCTGACCT cele-001-065.GCCATCAACT cele-001-068.ACGGCAACCA
##          "0.0"           "0.0"           "0.0"
## cele-001-071.GTCATTGCGC
##          "0.0"

```

Generate matrix of gene short names for Monocle

```

geneNames <- matrix(rownames(RawCountsPseudotime), ncol = 1)
rownames(geneNames) <- rownames(RawCountsPseudotime)
colnames(geneNames) <- "gene_short_name"
head(geneNames)

##      gene_short_name
## aap-1 "aap-1"
## aat-1 "aat-1"
## aat-2 "aat-2"
## aat-3 "aat-3"
## aat-4 "aat-4"
## aat-5 "aat-5"

```

Initiate Monocle object

```

cds <- new_cell_data_set(expression_data = RawCountsPseudotime,
                         cell_metadata = DataCluster.ID,
                         gene_metadata = geneNames)

cds <- cds[,names(ClusterPseudotime[ClusterPseudotime %in% c(
  "0.0",
  "0.1",
  # "1.0",
  # "1.1",
  "2.0"
  # "3.0",
  # "3.1",
  # "4.0",
  # "4.1",
  # "16.0",
  # "16.1",
  # "17.0",
  # "17.1",
  # "8.0.0",
  # "8.0.1",
  # "8.1",
  # "8.2",
  # "8.3",
  # "10.0.0",
  # "10.0.1_4",
  # "10.0.2_3",
  # "10.0.5",
  # "10.1.0",
  # "10.1.1",
  # "10.2",
  # "33.0",
  # "42.0",
  # "42.1"
  # "56.0"
)])] #all data for trajectories

```

```
colData(cds)
```

```

## DataFrame with 834 rows and 2 columns
##           Cluster.IDs      Size_Factor
##             <factor>        <numeric>
## cele-001-008.GATCAGTCAT 0.0 0.958473676933979
## cele-001-027.ACTCCGCCAA 0.0 0.814477631573005
## cele-001-042.TTCCTAGACC 0.0 1.29596440824876
## cele-001-046.TTCTACGCCA 0.0 0.908975036341144
## cele-001-047.TTCGCTGCCT 0.0 0.886475654253492
## ...
## cele-010-080.ATAGTCGTAG 2.0 0.611983192784137
## cele-010-085.TTCATAATGG 2.0 0.773978743815232
## cele-010-086.GTCGAGGTAT 2.0 0.494986405928346
## cele-010-090.AACTGATCTT 2.0 1.48045934136751
## cele-010-090.GGTAGATCAA 2.0 0.490486529510815

```

```

## Column Cluster.IDs contains the original DataCluster IDs
colData(cds)$Cluster.IDs <- factor(colData(cds)$Cluster.IDs)
colData(cds)

## DataFrame with 834 rows and 2 columns
##           Cluster.IDs      Size_Factor
##           <factor>       <numeric>
## cele-001-008.GATCAGTCAT    0.0 0.958473676933979
## cele-001-027.ACTCCGCAA    0.0 0.814477631573005
## cele-001-042.TTCCTAGACC   0.0 1.29596440824876
## cele-001-046.TTCTACGCCA   0.0 0.908975036341144
## cele-001-047.TTCGCTGCCT   0.0 0.886475654253492
## ...
## cele-010-080.ATAGTCGTAG   2.0 0.611983192784137
## cele-010-085.TTCATAATGG   2.0 0.773978743815232
## cele-010-086.GTCGAGGTAT   2.0 0.494986405928346
## cele-010-090.AACTGATCTT   2.0 1.48045934136751
## cele-010-090.GGTAGATCAA  2.0 0.490486529510815

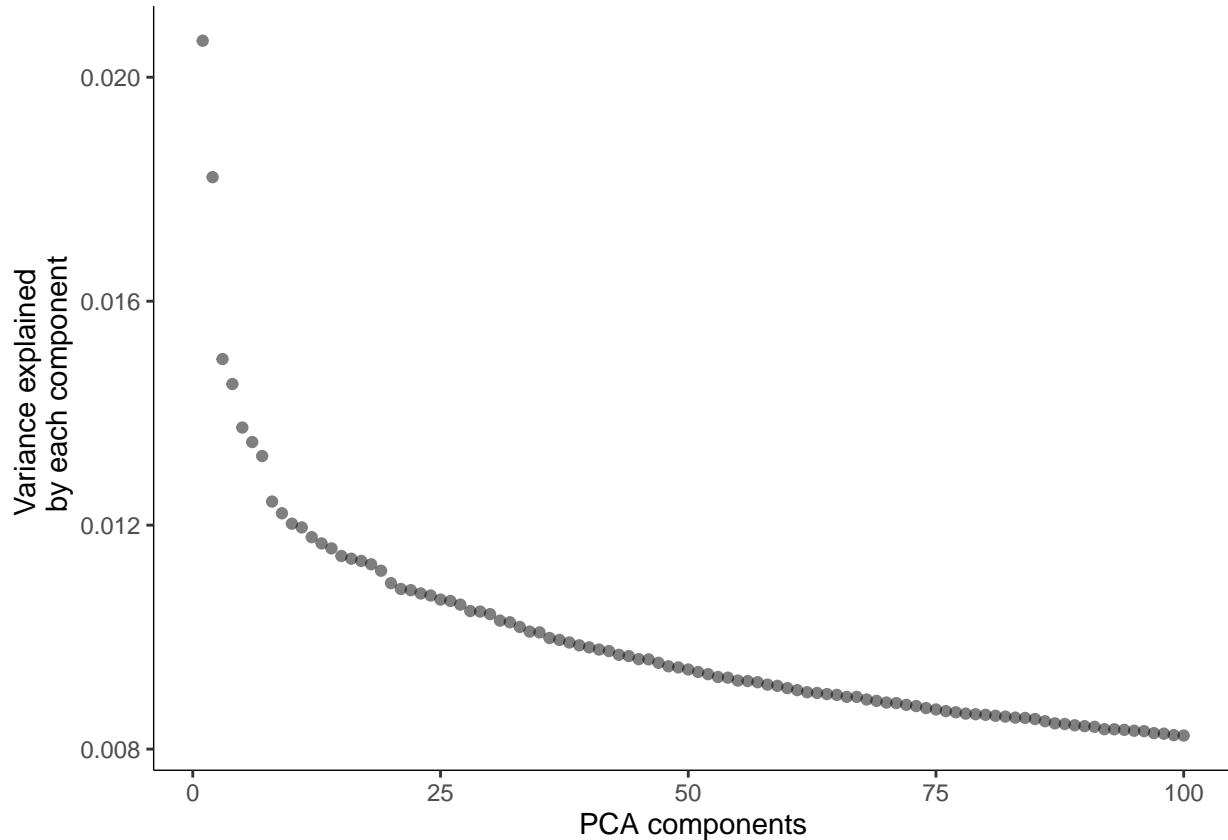
```

Step 1: Normalize and pre-process the data

```

cds <- preprocess_cds(cds, num_dim = 100)
plot_pc_variance_explained(cds)

```



```

cds <- preprocess_cds(cds, num_dim = 10)

```

Step 2: Reduce the dimensions using UMAP

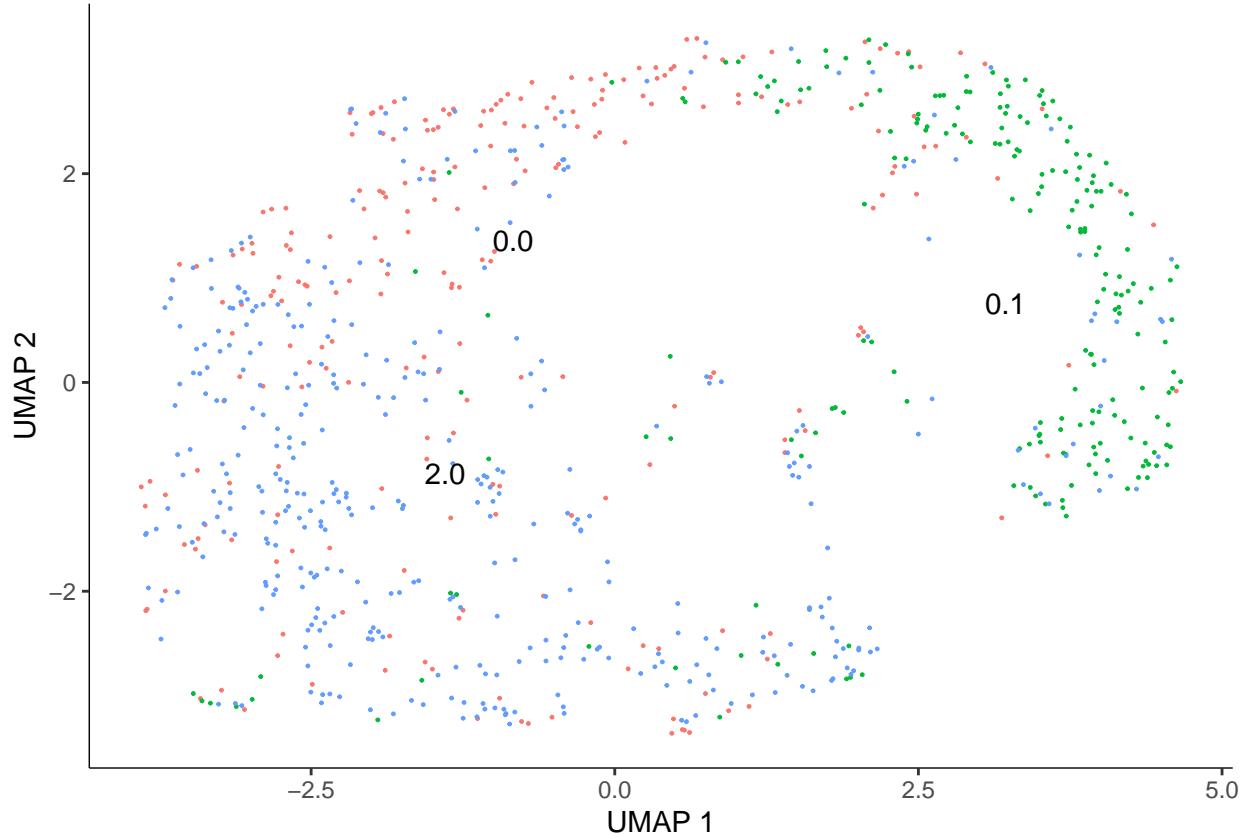
```
cds <- reduce_dimension(cds,cores = 8)

## No preprocess_method specified, using preprocess_method = 'PCA'

## Note: reduce_dimension will produce slightly different output each time you run it unless you set 'un

plot_cells(cds, color_cells_by = "Cluster.IDs", group_label_size = 4, cell_size = 0.5,show_trajectory_g

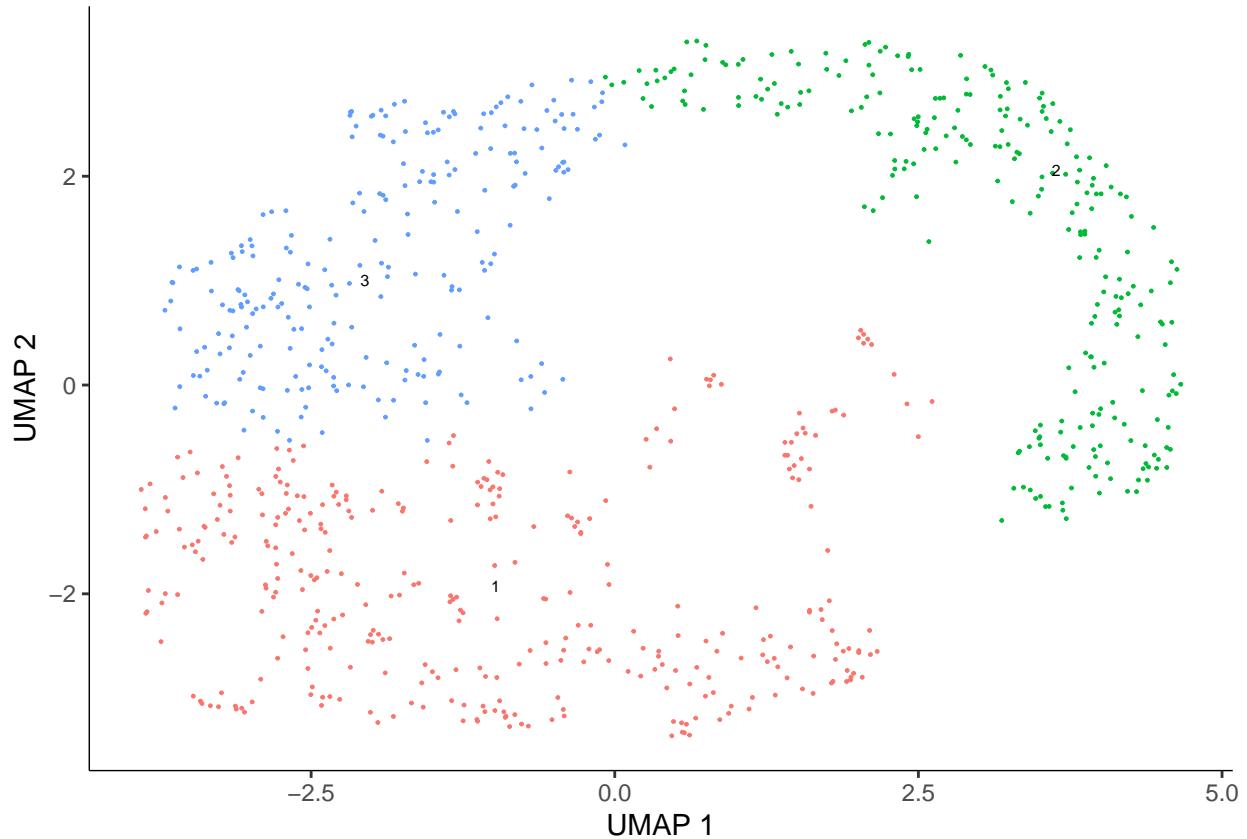
## No trajectory to plot. Has learn_graph() been called yet?
```



Step 3: Cluster the cells

```
# cds = cluster_cells(cds, resolution=0.02)
cds = cluster_cells(cds, resolution=0.004)
plot_cells(cds, cell_size = 0.5)

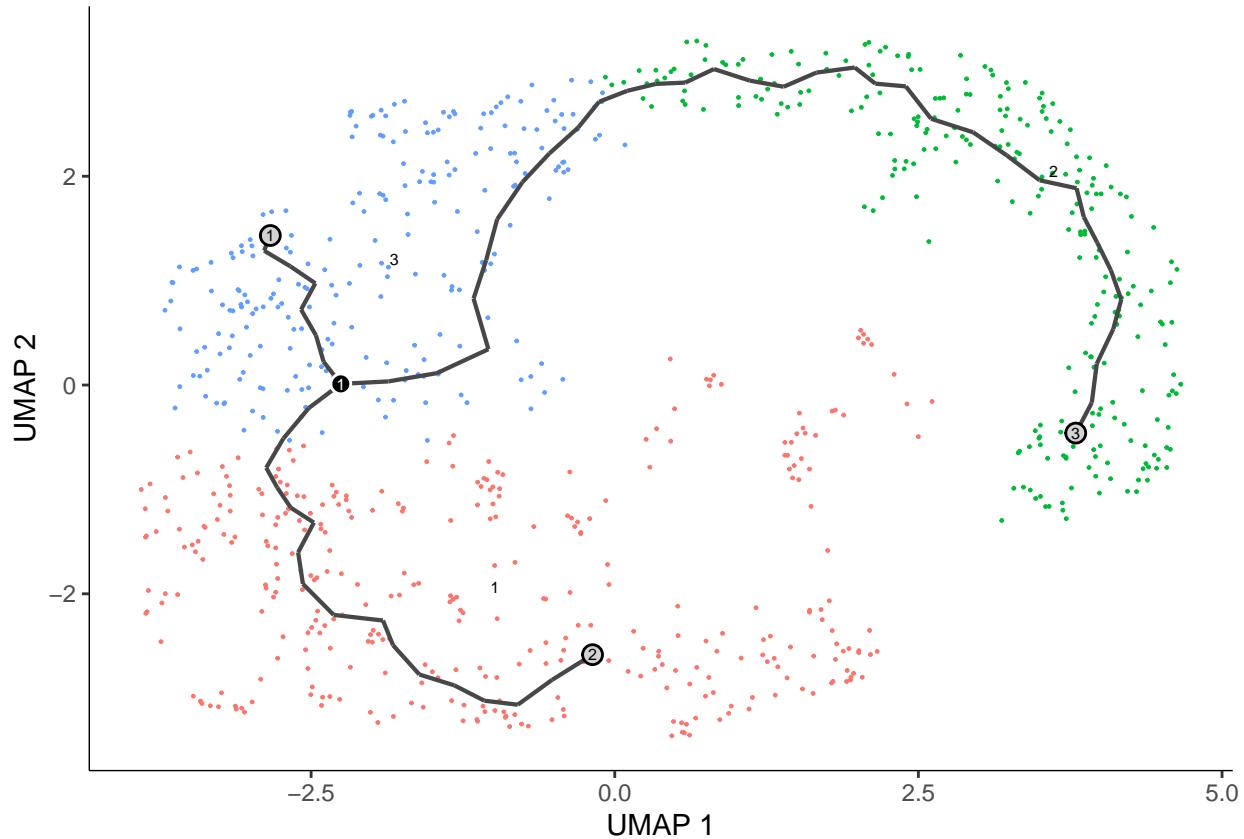
## No trajectory to plot. Has learn_graph() been called yet?
```



Step 4: Learn a graph and order cells

```
cds <- learn_graph(cds)

##  
| | 0%  
| |=====| 100%  
plot_cells(cds, cell_size = 0.5)
```



```

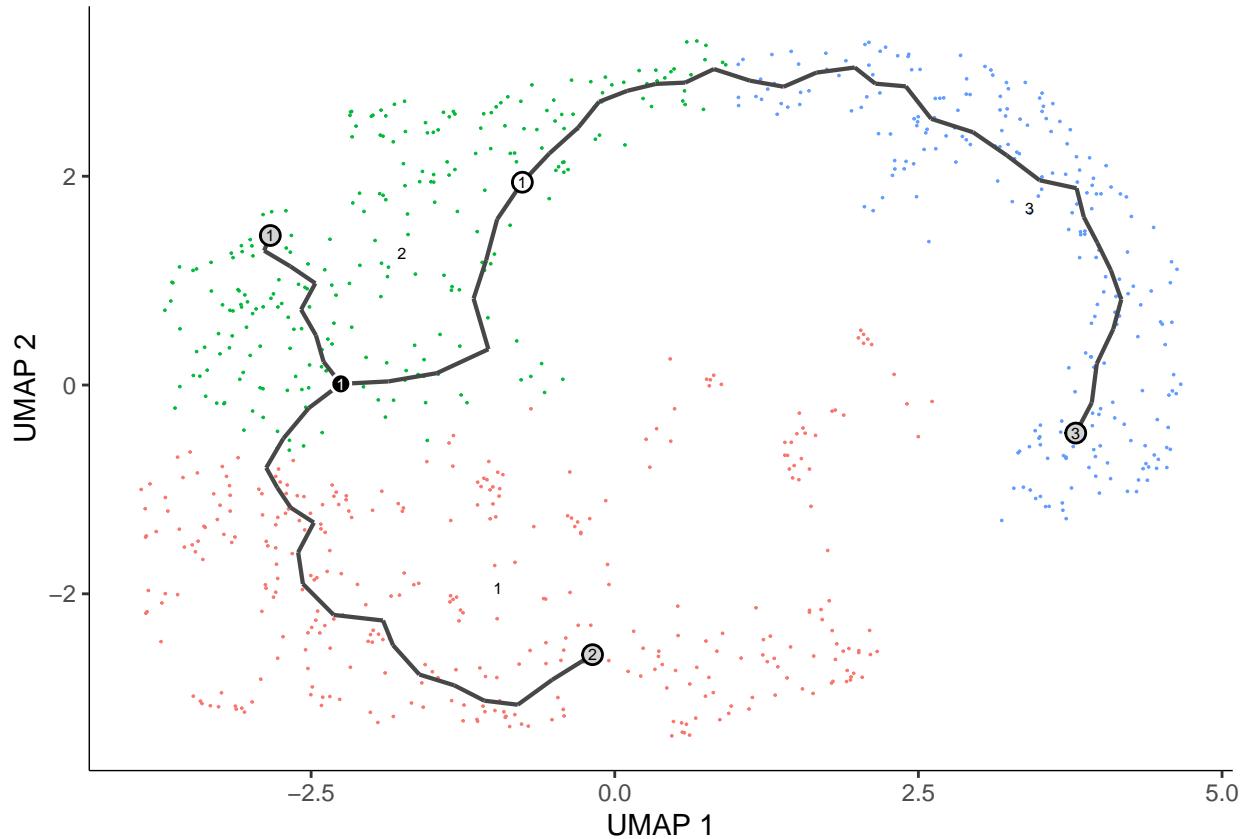
# ## With Shiny
# cds <- order_cells(cds)
# #
# save(file = "cdsRoots.rda", cds) #Save object state after selection of roots to avoid changes in furu
# cds_sub0.1 <- choose_graph_segments(cds)
# save(file = "cds_sub0.1.rda", cds_sub0.1)
#
# cds_sub2.0 <- choose_graph_segments(cds)
# save(file = "cds_sub2.0.rda", cds_sub2.0)

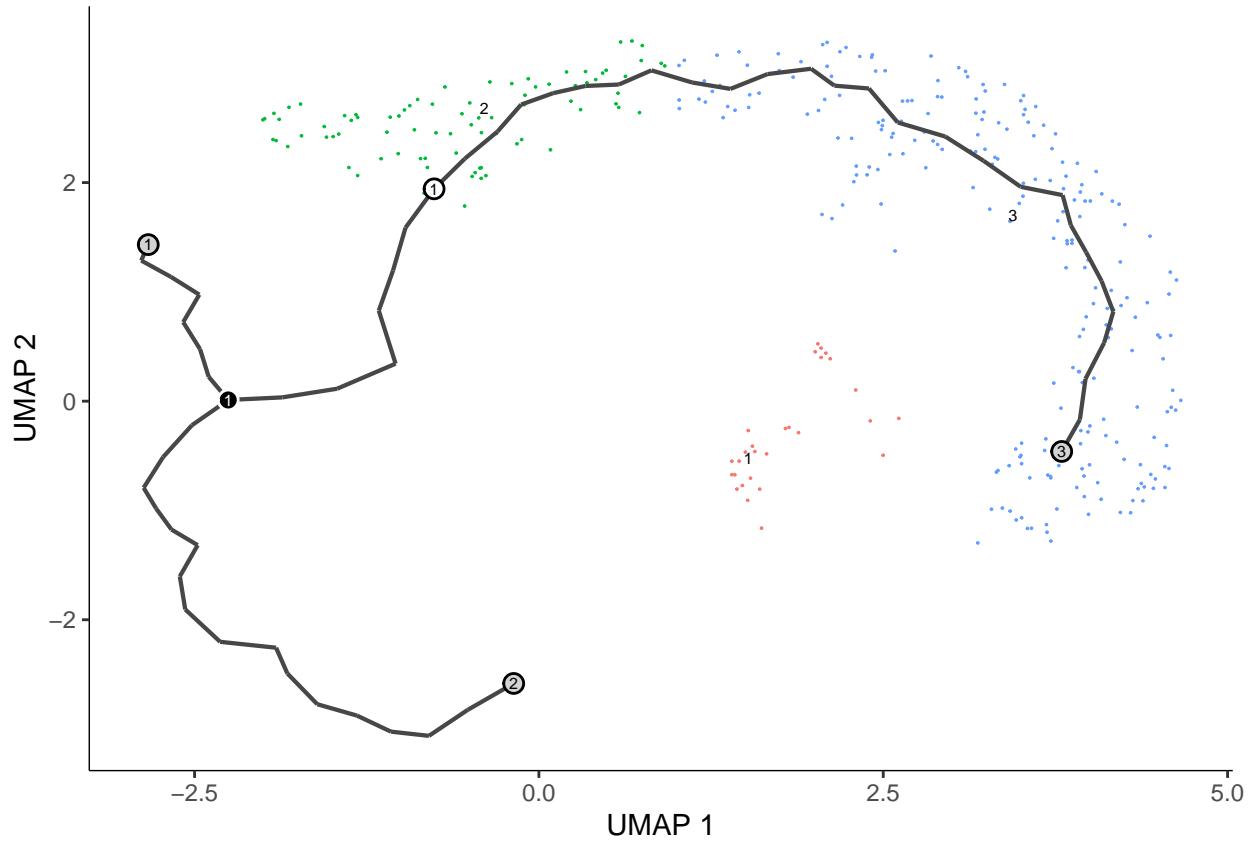
## Without Shiny
load("cdsRoots.rda") #Load object
load("cds_sub0.1.rda")
load("cds_sub2.0.rda")

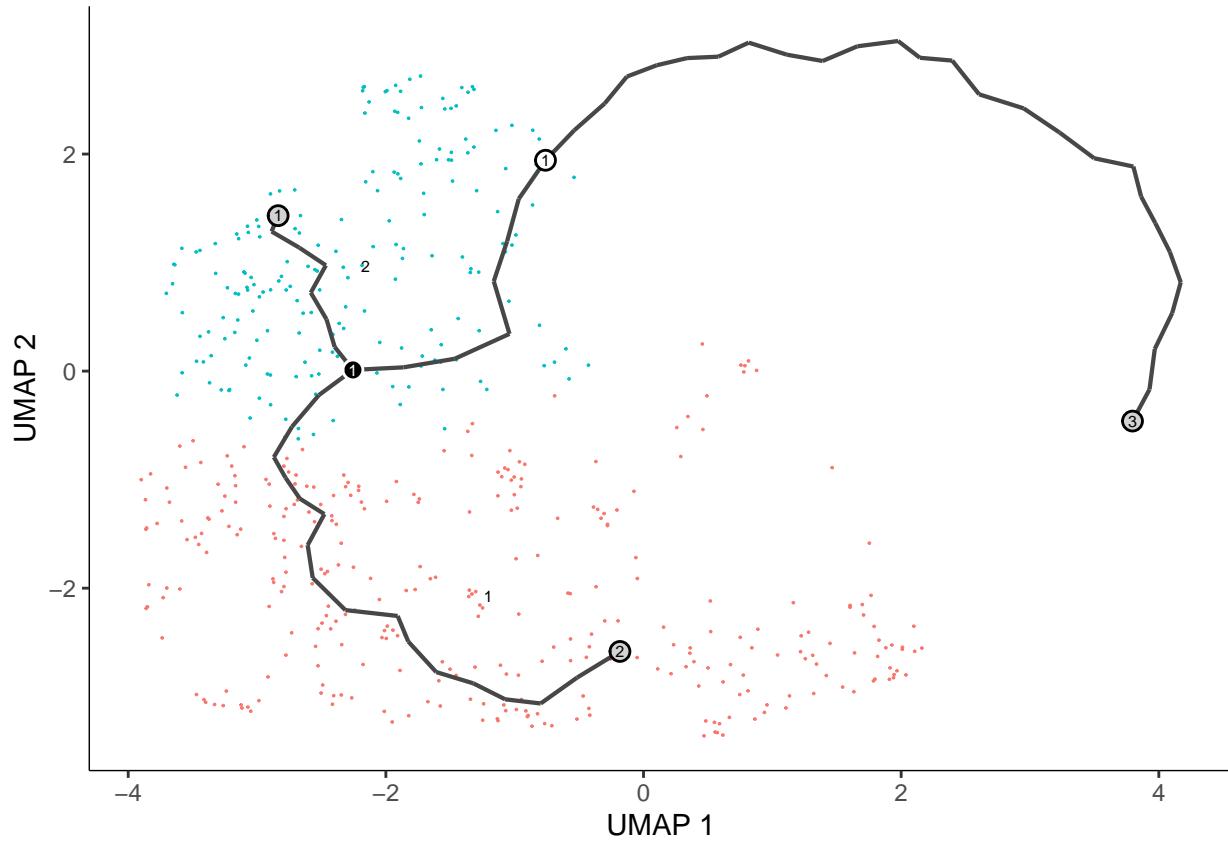
print(cds@principal_graph_aux$UMAP$root_pr_nodes)

## [1] "Y_4"
plot_cells(cds)

```

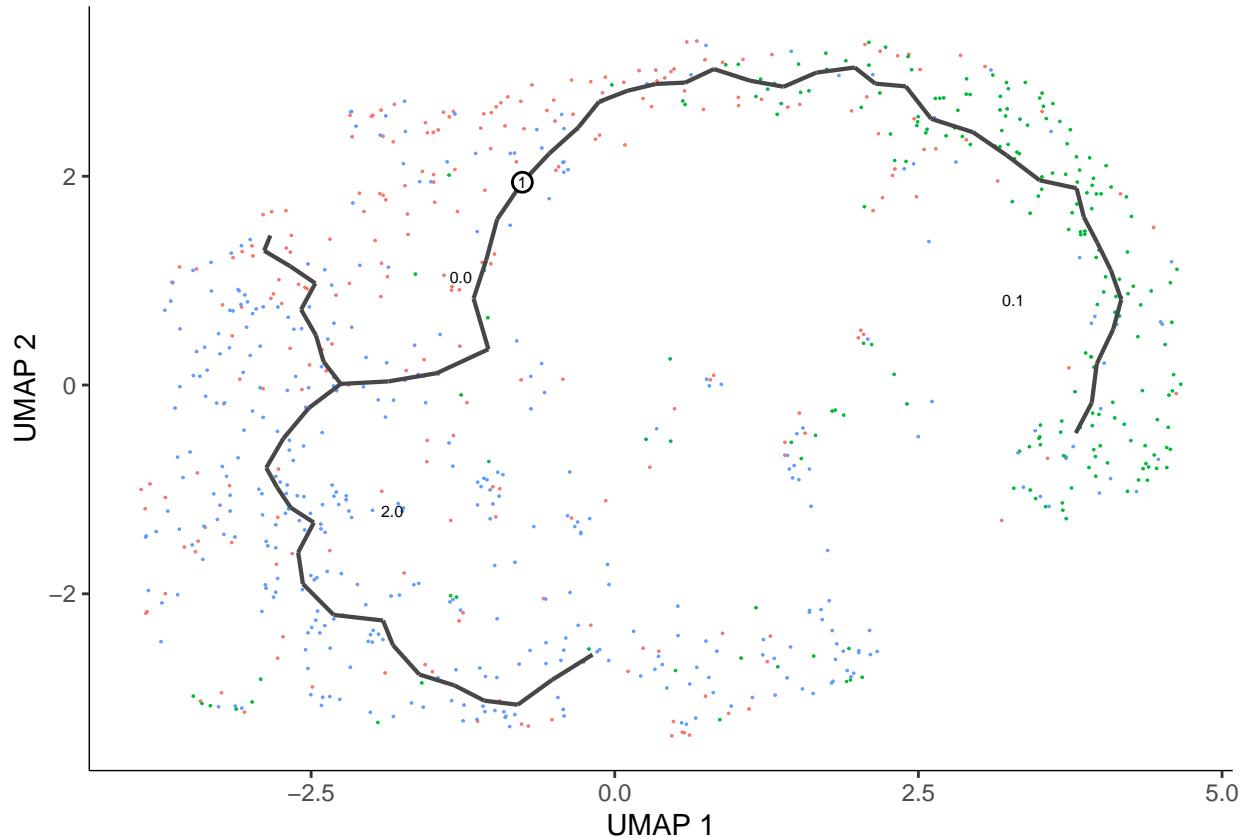






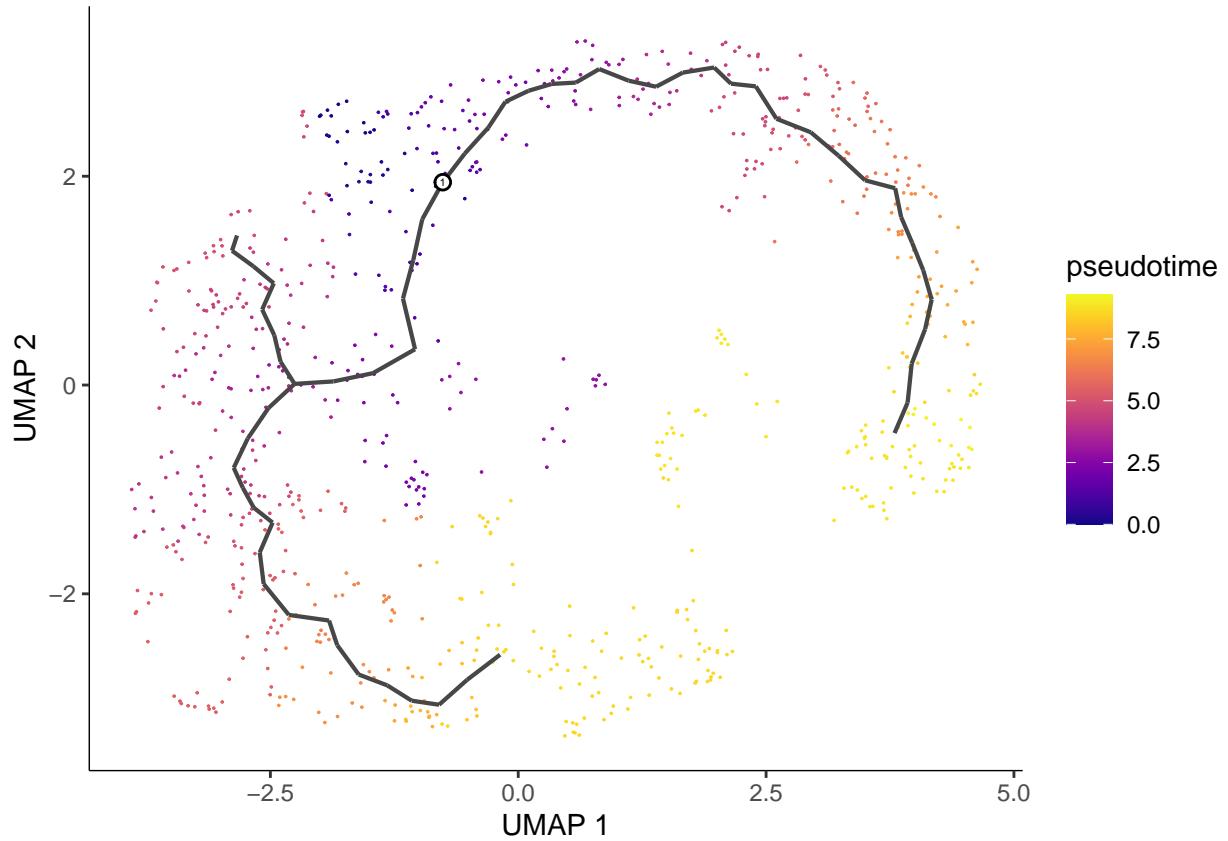
Plot the pseudotime graph

```
plot_cells(cds,
            color_cells_by = "Cluster.IDs",
            label_groups_by_cluster=FALSE,
            label_leaves=FALSE,
            label_branch_points=FALSE)
```

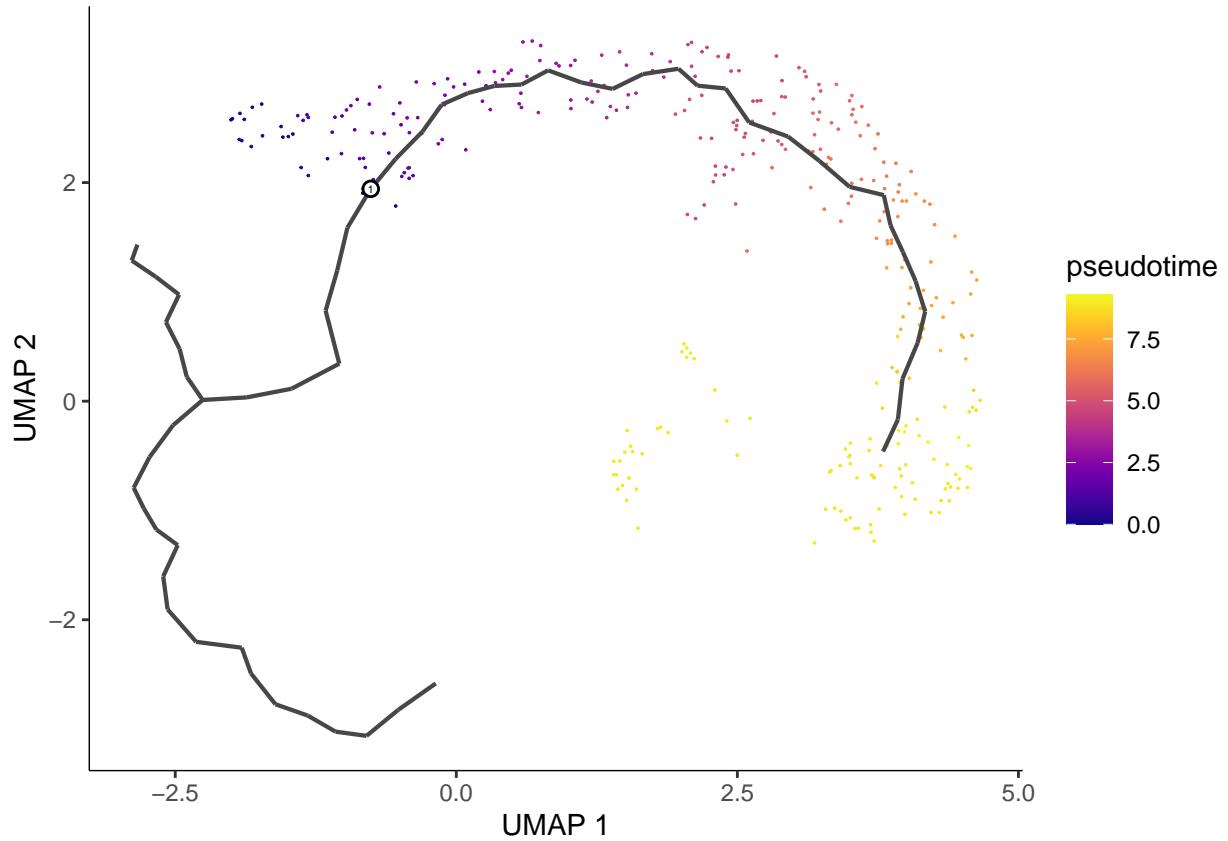


Plot the pseudotime graph

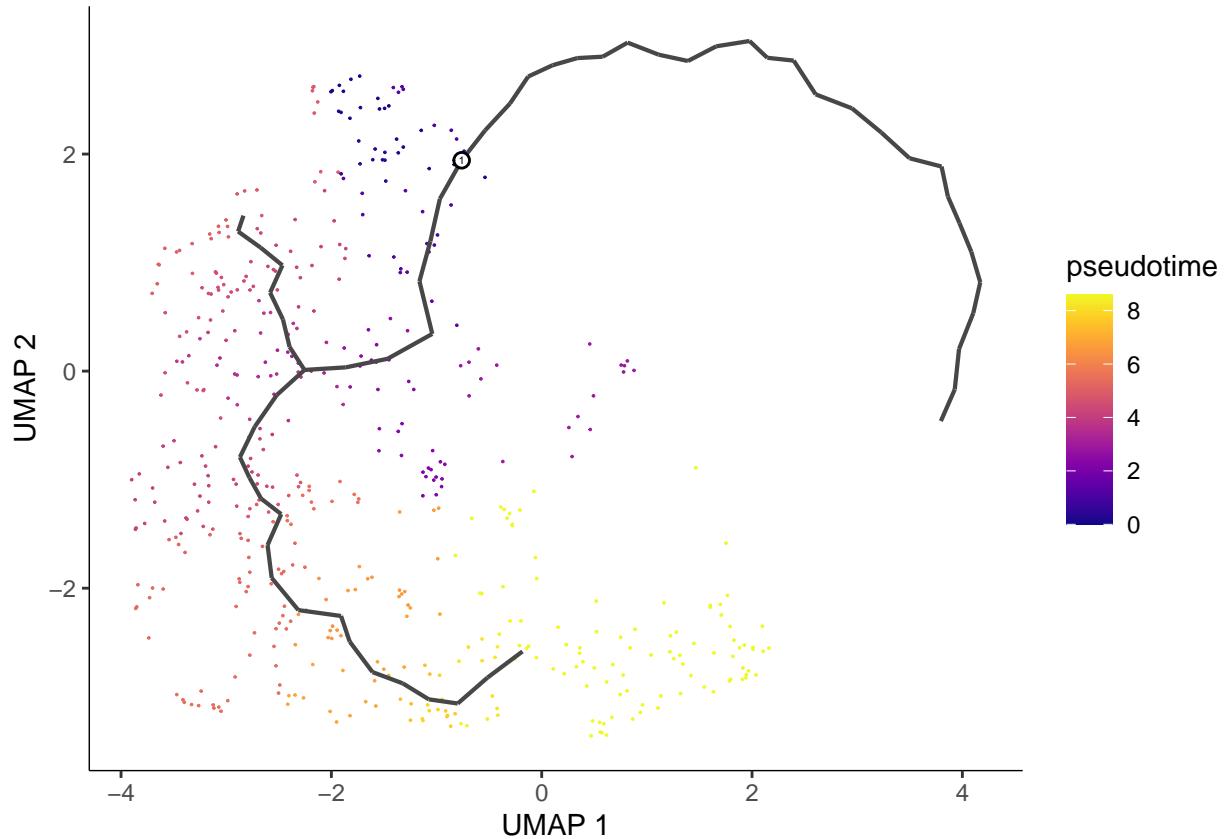
```
plot_cells(cds,
           color_cells_by = "pseudotime",
           label_cell_groups=FALSE,
           label_leaves=FALSE,
           label_branch_points=FALSE,
           graph_label_size=1.5)
```



```
plot_cells(cds_sub0.1,
           color_cells_by = "pseudotime",
           label_cell_groups=FALSE,
           label_leaves=FALSE,
           label_branch_points=FALSE,
           graph_label_size=1.5)
```

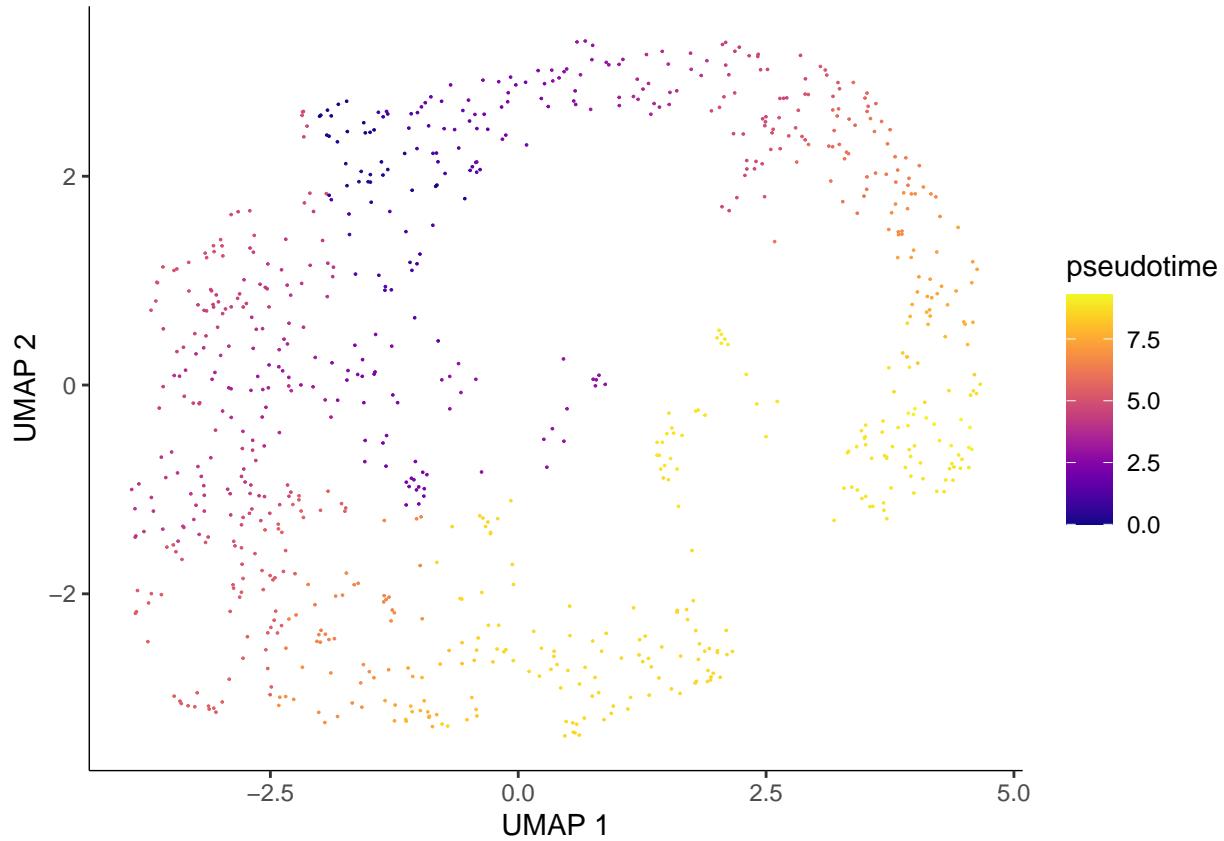


```
plot_cells(cds_sub2.0,
           color_cells_by = "pseudotime",
           label_cell_groups=FALSE,
           label_leaves=FALSE,
           label_branch_points=FALSE,
           graph_label_size=1.5)
```



Plot the pseudotime graph without trajectory

```
plot_cells(cds,
            color_cells_by = "pseudotime",
            label_cell_groups=FALSE,
            label_leaves=FALSE,
            label_branch_points=FALSE,
            graph_label_size=1.5,
            show_trajectory_graph = FALSE)
```



Correlation Analysis

```

cds_pg <- graph_test(cds, neighbor_graph="principal_graph", cores=4, verbose = F)

cds_genes <- cds_pg %>%
  filter(q_value < 0.05) %>%
  arrange(desc(morans_I)) %>%
  select(gene_short_name)

cds_genes$gene_short_name

##    [1] ftt-2      C56G2.4    F35H12.6   unc-6      rps-8      Y44A6D.2
##    [7] aldo-1     enol-1     rla-0       rla-1      eef-1A.1   rpl-4
##   [13] rpl-21     flp-9       his-72     gpb-1      mua-3      pqn-52
##   [19] rps-29     Y37E3.8    rpl-13     icln-1     aqp-2      parn-2
##   [25] rpl-22     rps-0       oaz-1      eef-1A.2   K11H12.7  cmd-1
##   [31] ret-1      rpl-33     F09C3.2    aak-1      H36L18.2  anat-1
##   [37] twk-49     rpl-18     Y92H12BR.2 rpl-31     rps-10     ZC21.10
##   [43] gdi-1      rps-14     rpl-23     ufbp-1     rpl-10     tim-1
##   [49] W01D2.1    rab-3      rpl-15     rps-13     ric-4      eef-1B.1
##   [55] F13E9.11   sms-2      kin-2      ubl-1      ant-1.1   Y82E9BR.3
##   [61] cyn-7      rack-1     pat-6      vha-4      rpl-7      rpl-5
##   [67] gpd-2      T19H5.4    rpl-19     rpl-32     ben-1      rps-15
##   [73] unc-80     rps-3      rps-6      dpy-11     fhod-1     rpl-12
##   [79] B0228.6    cars-1     egl-3      zyx-1      F10G8.2    rpl-24.1
##   [85] nlp-38     let-92     gar-2      par-5      K07C11.7  K09F6.13

```

## [91]	T25B9.8	T01G1.2	tag-80	flp-14	apl-1	nep-21
## [97]	rpl-27	spe-39	mks-3	acly-1	Y82E9BR.19	ptp-4
## [103]	rpl-30	gei-4	rps-9	rpl-11.2	his-24	casy-1
## [109]	Y97E10C.1	rps-19	ZK1098.11	cct-5	rpl-35	cyh-1
## [115]	memo-1	snt-4	C02B10.3	sig-7	tbcb-1	rps-27
## [121]	C06A1.2	rbc-2	rps-28	tba-1	ehs-1	athp-2
## [127]	rps-18	dsl-3	F10G8.8	ugt-64	pck-1	swt-1
## [133]	frpr-5	act-4	rps-5	T23G7.2	mpz-1	bed-3
## [139]	gpd-3	R119.3	let-70	Y106G6D.5	unc-17	bar-1
## [145]	helq-1	lem-4	ceh-6	dpy-28	Y55F3AM.3	Y57E12AL.6
## [151]	par-6	vha-8	F11A10.6	F02A9.4	mdt-20	acbp-1
## [157]	K07C5.3	madf-2	unc-108	tpi-1	rpc-1	unc-53
## [163]	R07E5.1	C17H12.2	sem-4	pgk-1	rps-11	B0336.3
## [169]	rpl-16	pab-1	F54E4.3	glt-1	B0198.3	cey-1
## [175]	mxl-3	hda-4	tbb-2	rpl-36	mkk-4	cdh-1
## [181]	ZC434.9	F21D5.9	rpl-1	fmi-1	sec-8	mig-23
## [187]	F54C9.11	rps-2	C04H5.1	F46H5.3	C37A2.7	vdac-1
## [193]	R09A8.5	rpl-3	ZC15.10	B0511.13	apc-11	rpl-7A
## [199]	mrps-12	C41D11.3	abu-8	chch-3	Y54G2A.26	far-1
## [205]	gpc-2	ZK686.2	dars-1	trap-1	twk-17	tct-1
## [211]	vav-1	rps-22	Y46H3C.7	maph-1.1	hlh-32	rhgf-1
## [217]	Y106G6G.6	eef-2	C30A5.3	cpx-1	Y105C5B.19	ndk-1
## [223]	ergo-1	Y43F8B.2	tsp-11	C45G9.5	arl-8	Y16B4A.2
## [229]	T27C4.1	acl-4	set-20	rps-25	rpl-34	D2030.3
## [235]	T12E12.3	paa-1	atp-5	F36A2.7	hil-3	daf-21
## [241]	unc-13	acr-9	M163.1	C50D2.7	goa-1	C29F9.1
## [247]	C08G5.7	nlf-1	F32D1.5	clp-1	tat-3	ceh-21
## [253]	tre-1	snb-1	Y47D3B.6	egl-30	Y57G11C.3	ZK484.4
## [259]	ile-1	cogc-2	T19C3.3	F02E9.5	crb-1	lgg-1
## [265]	Y43F4B.7	W05H12.1	F57A8.1	M04B2.7	R02F2.1	rps-4
## [271]	C23H3.5	Y48C3A.20	rpl-25.1	clp-7	lin-46	nuo-3
## [277]	rpl-6	deg-1	ZK652.6	cyp-33C8	lnkn-1	rpl-28
## [283]	T05H10.8	hyls-1	F48G7.4	marb-1	C35B1.5	unc-9
## [289]	sqv-1	Y53F4B.14	eef-1G	pfk-1.1	lat-1	C35B8.3
## [295]	F52C12.2	Y57A10A.10	C47E12.2	adt-1	asb-2	oxi-1
## [301]	nhr-40	arf-3	mel-32	phf-10	K05F1.6	Y57G11C.46
## [307]	rps-23	tofu-5	rpl-17	ufl-1	rps-26	nduo-5
## [313]	rmo-1	T26A5.8	bath-42	test-1	ubq-2	pqn-85
## [319]	skr-1	dmsr-3	unc-18	glr-5	atf-5	F10C2.3
## [325]	vps-51	ZK1248.11	tin-10	ldh-1	cfz-2	dtmk-1
## [331]	erp-1	sem-5	hmg-1.1	F26D11.1	ran-4	W09C5.7
## [337]	dbn-1	icd-1	vha-12	rps-16	nkb-1	unc-4
## [343]	iff-2	unc-64	cct-1	qns-1	skih-2	K08A2.4
## [349]	gsk-3	W08A12.3	K05C4.9	che-10	uri-1	Y48E1B.2
## [355]	K10C8.4	pde-4	lmn-1	madd-4	rpl-20	tam-1
## [361]	adbp-1	unc-104	F26E4.3	C15H9.9	cle-1	mec-15
## [367]	nep-26	rps-7	rpt-5	aqp-7	F58D5.5	Y106G6A.1
## [373]	ifg-1	hsp-1	F16B4.7	M18.3	rpl-43	cyc-1
## [379]	ran-5	nuo-6	unc-57	egl-47	F52H2.7	ckr-1
## [385]	cab-1	sng-1	F57B7.2	csk-1	trak-1	lgg-2
## [391]	ZK632.10	T07E3.4	F45E1.1	H34C03.2	C44E4.4	vha-15
## [397]	prp-19	tbb-1	rpl-26	H24K24.3	C56G7.3	dog-1
## [403]	Y18H1A.18	T01G1.4	T18D3.7	F25H2.12	F29B9.11	gon-1
## [409]	slc-17.1	ZK470.2	mrps-15	wip-1	T28D6.5	mdh-1

```

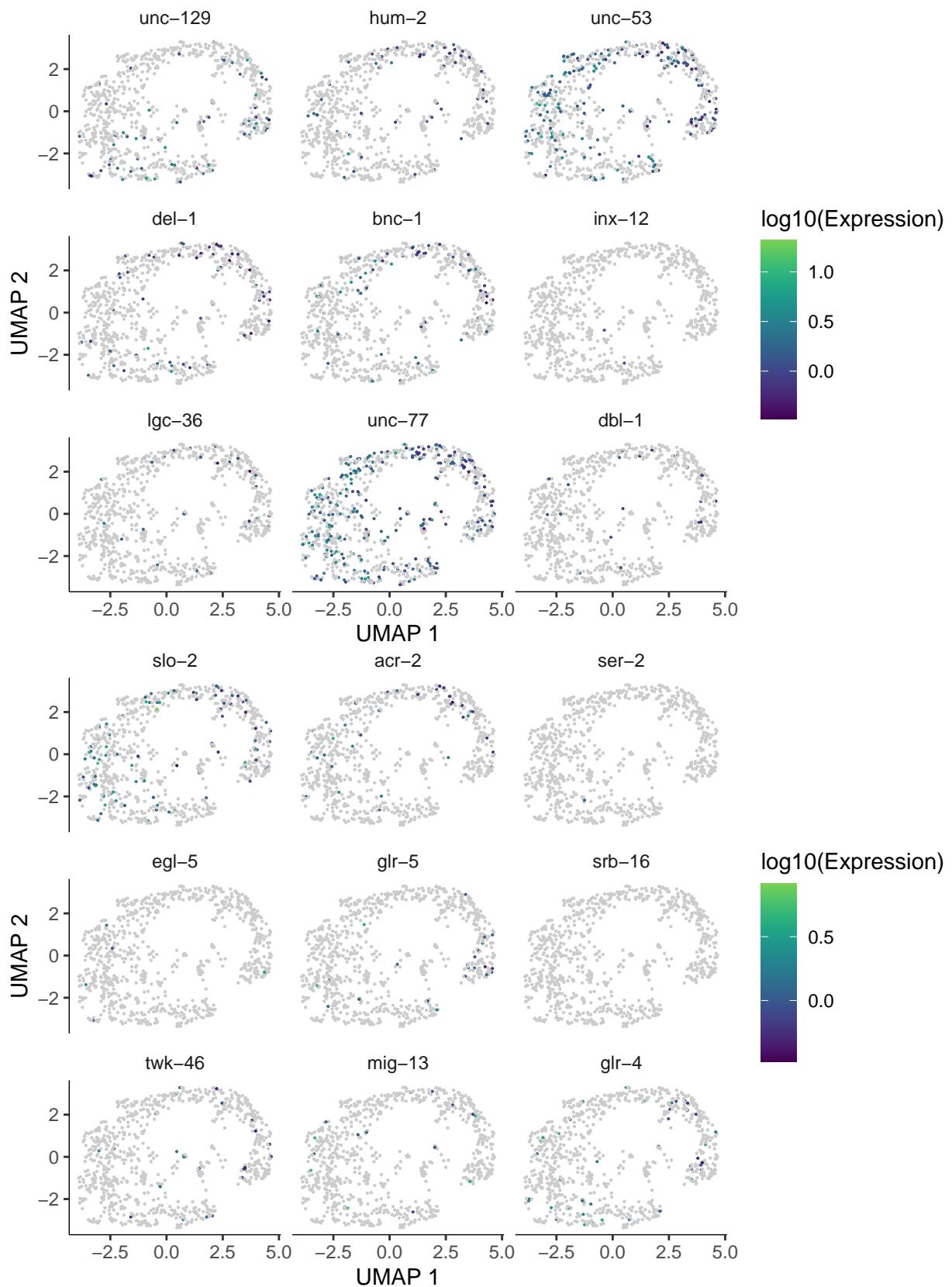
## [415] F08G5.3    mig-39    alh-7     F44E5.1    F59F5.5    atg-18
## [421] zer-1      nlp-14    spas-1    ptp-3      npp-6      trpp-5
## [427] unc-52     Y17G7B.22  atp-2     cutl-19   C24A3.2    hum-2
## [433] C46C2.2    sri-18    C50F7.6   nsf-1      Y53G8AR.8  C16H3.3
## [439] nkb-3      fum-1     flp-11    F26F12.3  K01A11.1   Y37E3.1
## [445] twk-29     arrd-22   nrx-1     wrk-1      F32D8.10   T24D5.2
## [451] C06E4.2    glb-13    sod-1     C24H12.4  ttb-1      tads-1
## [457] rsp-2      Y41D4A.4  mex-3     car-1      strd-1    F55D12.2
## [463] T04F8.3    B0001.5   rpy-1     Y82E9BR.18 C46F11.4  nipa-1
## [469] nep-11     F16A11.1  F07C3.2   epn-1      elt-1      map-2
## [475] pmr-1      nhr-14    B0454.9   cdc-42    nap-1      Y45F10D.10
## [481] ztf-3      W10C4.1   C23F12.4  pat-10    mrpl-50    Y105E8B.7
## [487] dnj-10     R02D3.4   C47D12.2  Y37E11B.5 R07E4.1   tag-72
## [493] nstp-10   wah-1     K09E9.1   acox-5    sna-1      mrpl-18
## [499] Y94H6A.3   R09F10.5  F40F11.3   spl-1      egl-13    R13H4.7
## [505] F07H5.8    arx-5     R01B10.6  C13F10.6  C06E8.5   T05F1.7
## [511] D1007.10   cyp-33C12 T02D1.8   F25E5.8   F13H10.8  K02E11.10
## 20271 Levels: 2L52.1 2RSSE.1 4R79.2 6R55.2 aagr-1 aagr-2 aagr-3 aagr-4 ... zyx-1

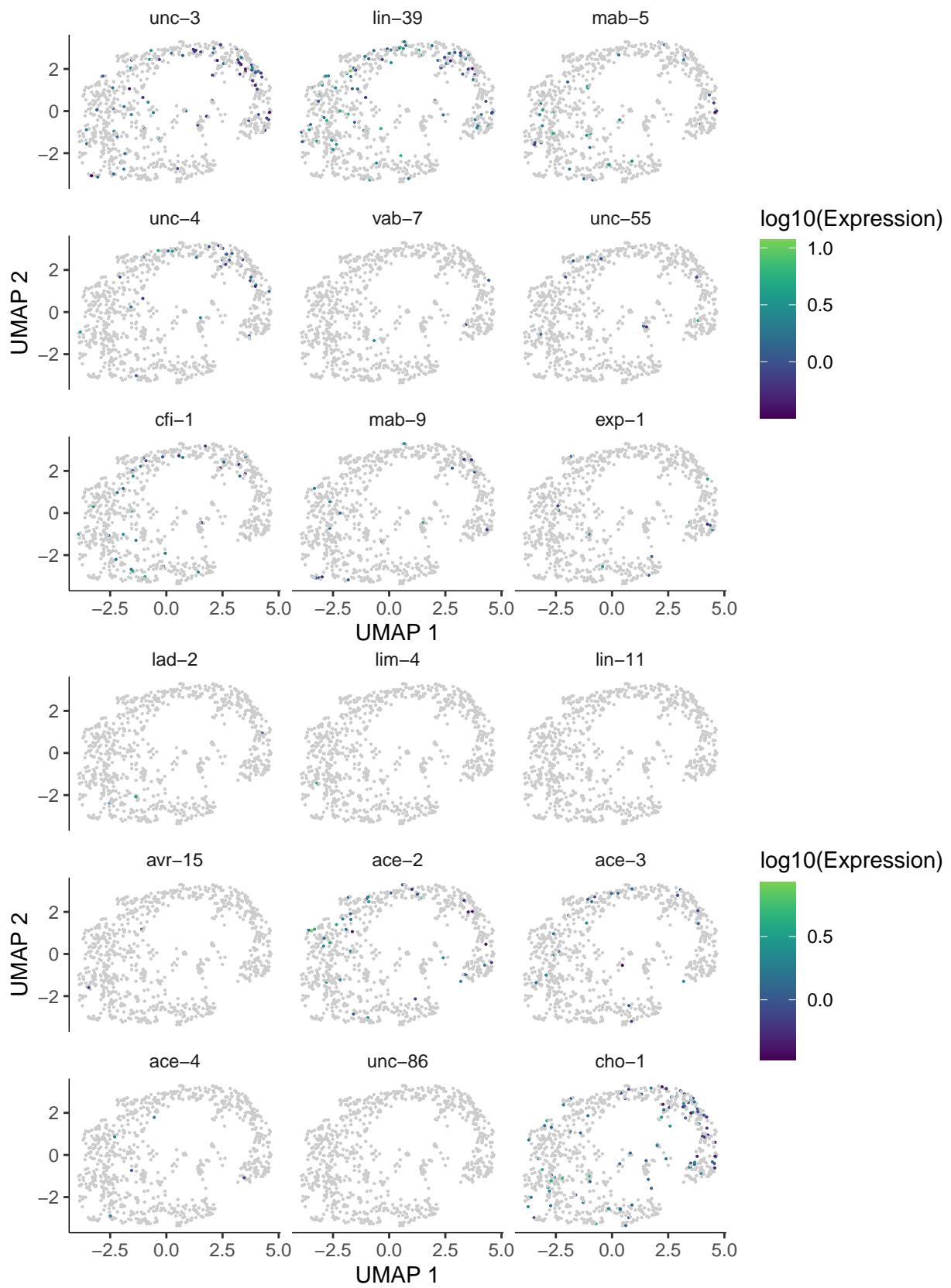
# Plot genes
list_genes <- unique(c("unc-129" , "hum-2", "unc-53", "del-1", "bnc-1", "inx-12" , "lgc-36", "unc-77

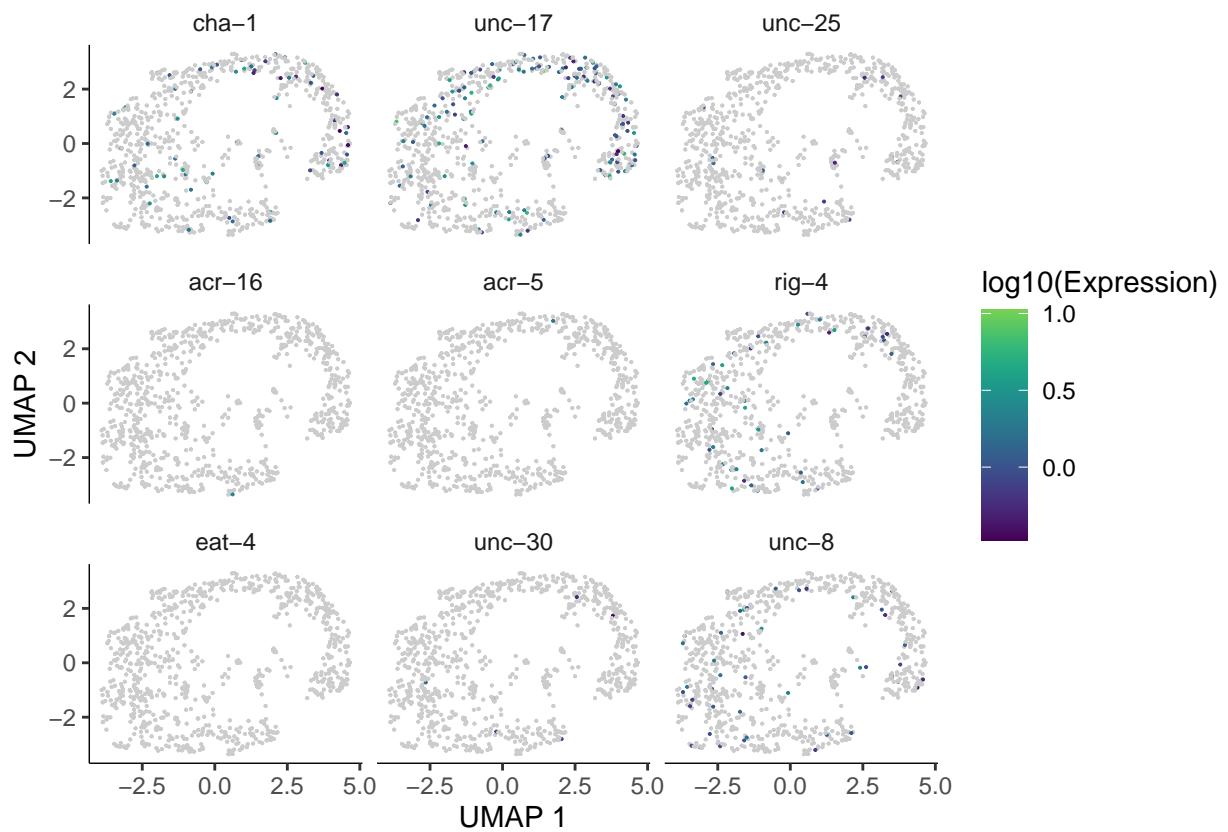
nplots <- 9
x <- seq_along(list_genes)
toplot <- split(list_genes, ceiling(x/nplots))
fillplot <- nplots-length(toplot[[length(toplot)]])
if (fillplot != 0) {
  toplot[[length(toplot)]] <- c(toplot[[length(toplot)]],toplot[[1]][1:fillplot])
}

for (x in toplot) {
  print(plot_cells(cds, genes= x,
                  show_trajectory_graph=FALSE,
                  label_cell_groups=FALSE,
                  label_leaves=FALSE,
                  cell_size = .4
                  ))
}

```







Pseudotemporal Expression Pattern

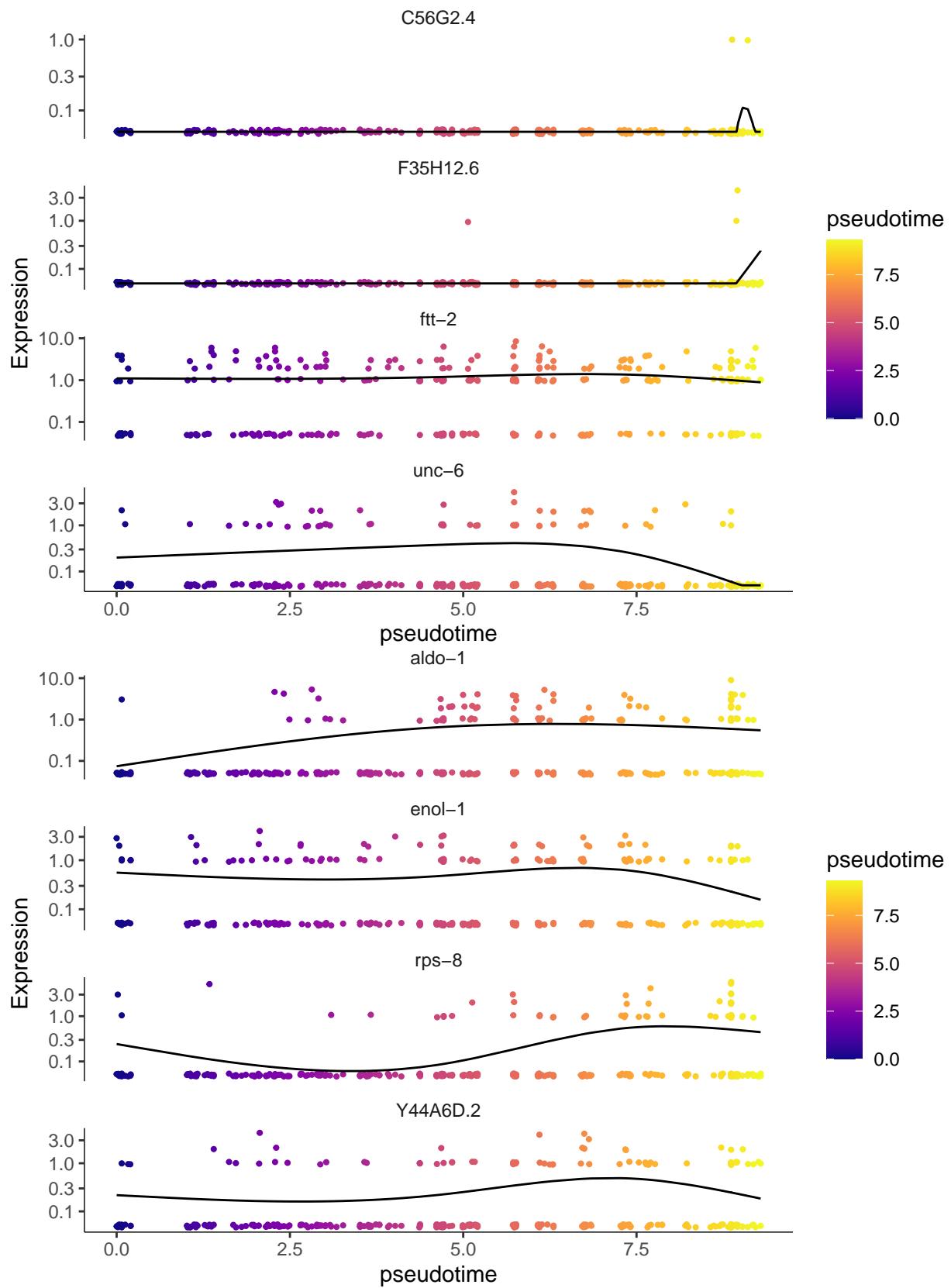
Differentially expressed cds_sub0.1

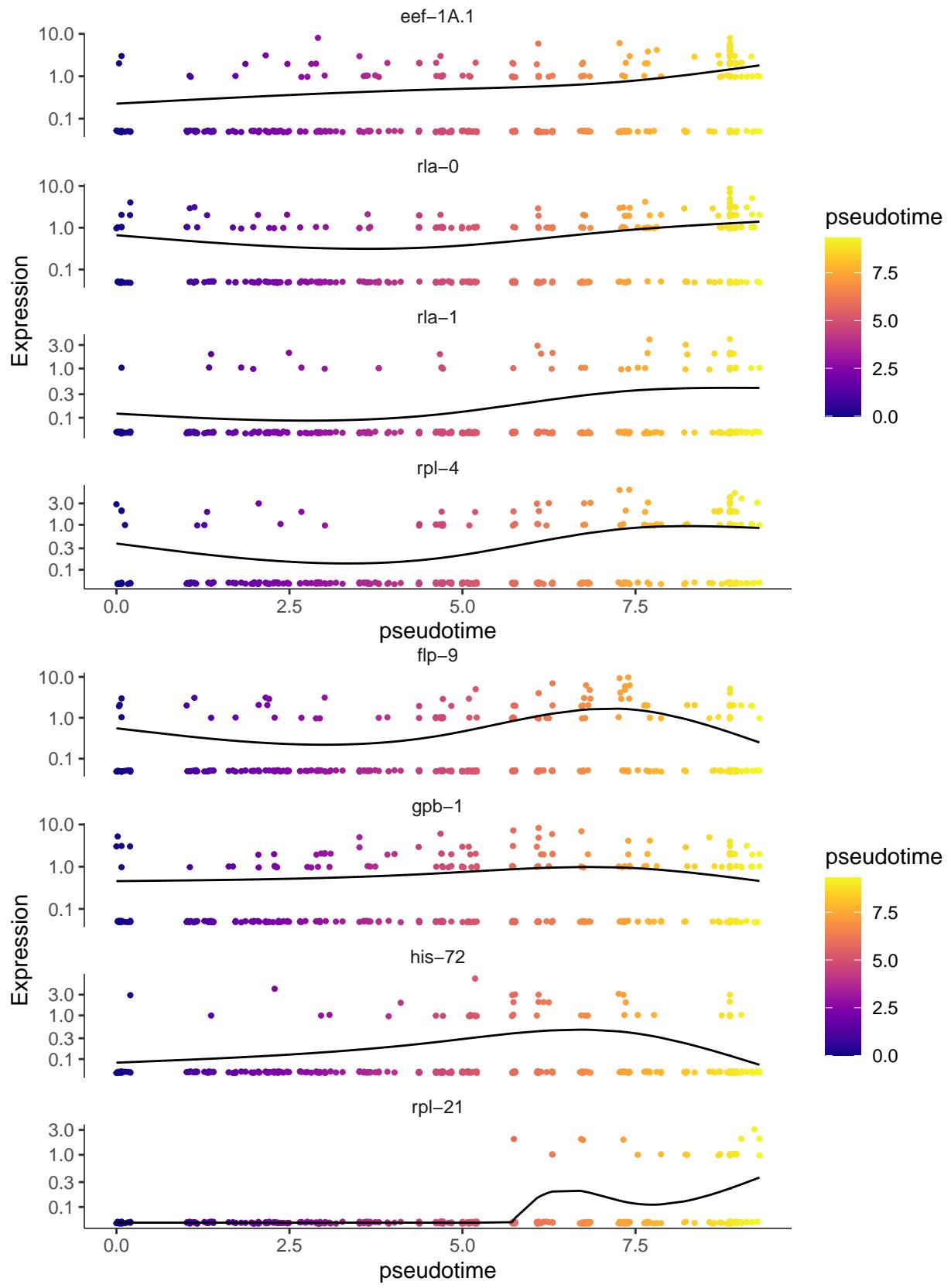
```

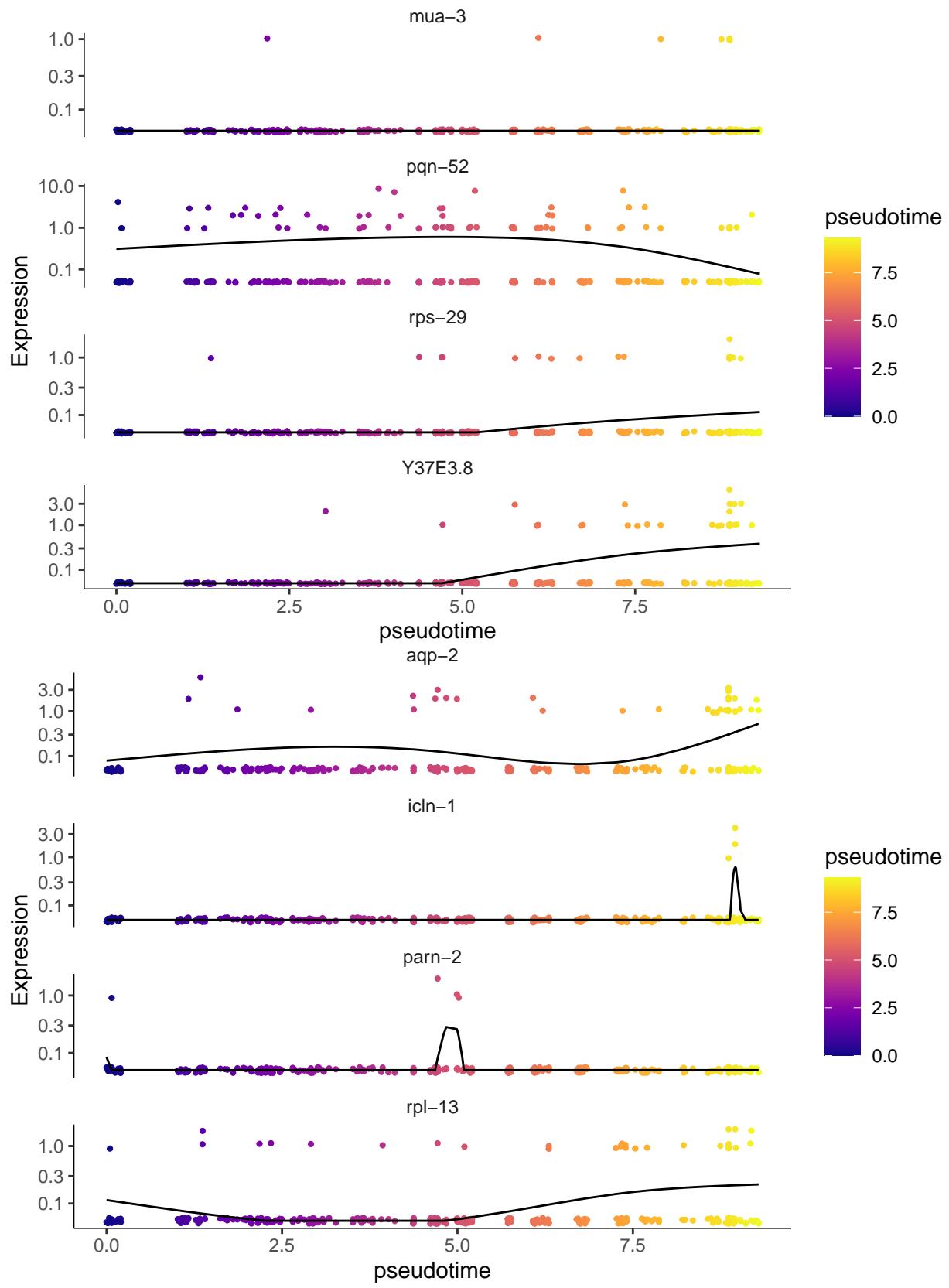
nplots <- 4
x <- seq_along(as.character(cds_genes$gene_short_name[1:40]))
toplot <- split(as.character(cds_genes$gene_short_name[1:40]), ceiling(x/nplots))
fillplot <- nplots-length(toplot[[length(toplot)]])
if (fillplot != 0) {
  toplot[[length(toplot)]] <- c(toplot[[length(toplot)]], toplot[[1]][1:fillplot])
}

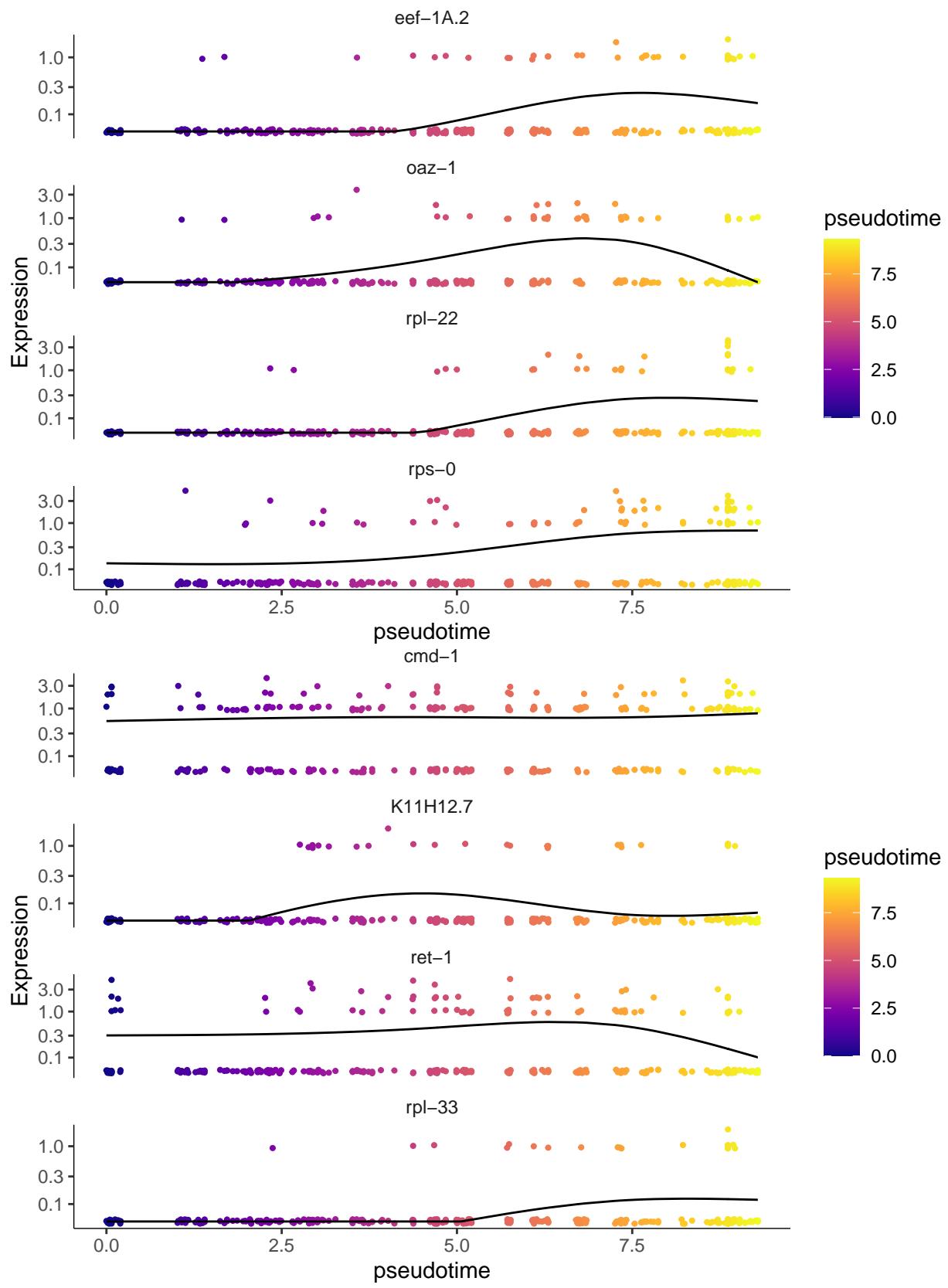
for (x in toplot) {
  cds_lineage_cds <- cds_sub0.1[rowData(cds_sub0.1)$gene_short_name %in% x,]
  print(plot_genes_in_pseudotime(cds_lineage_cds,
                                  min_expr=0.05))
}

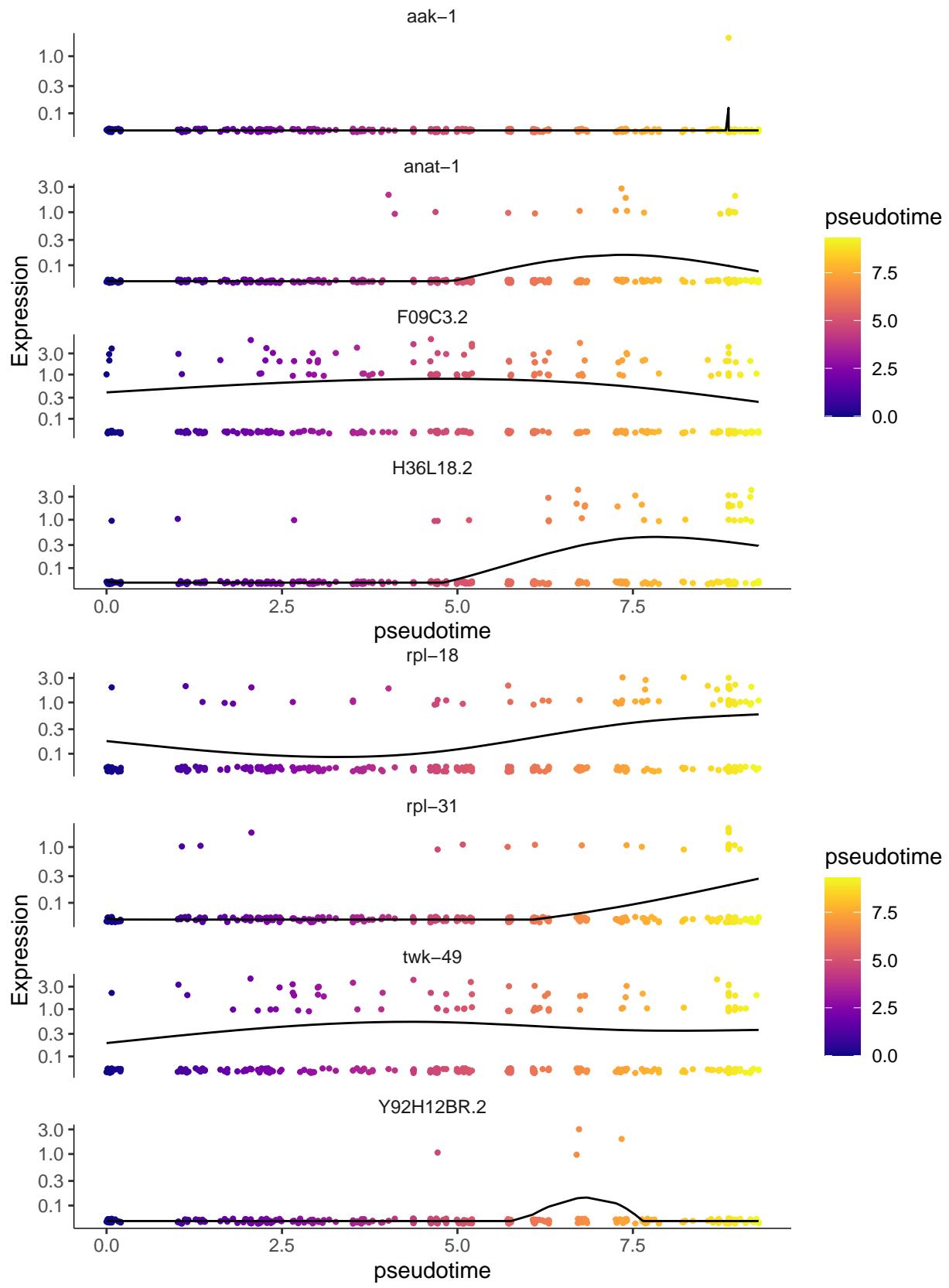
```











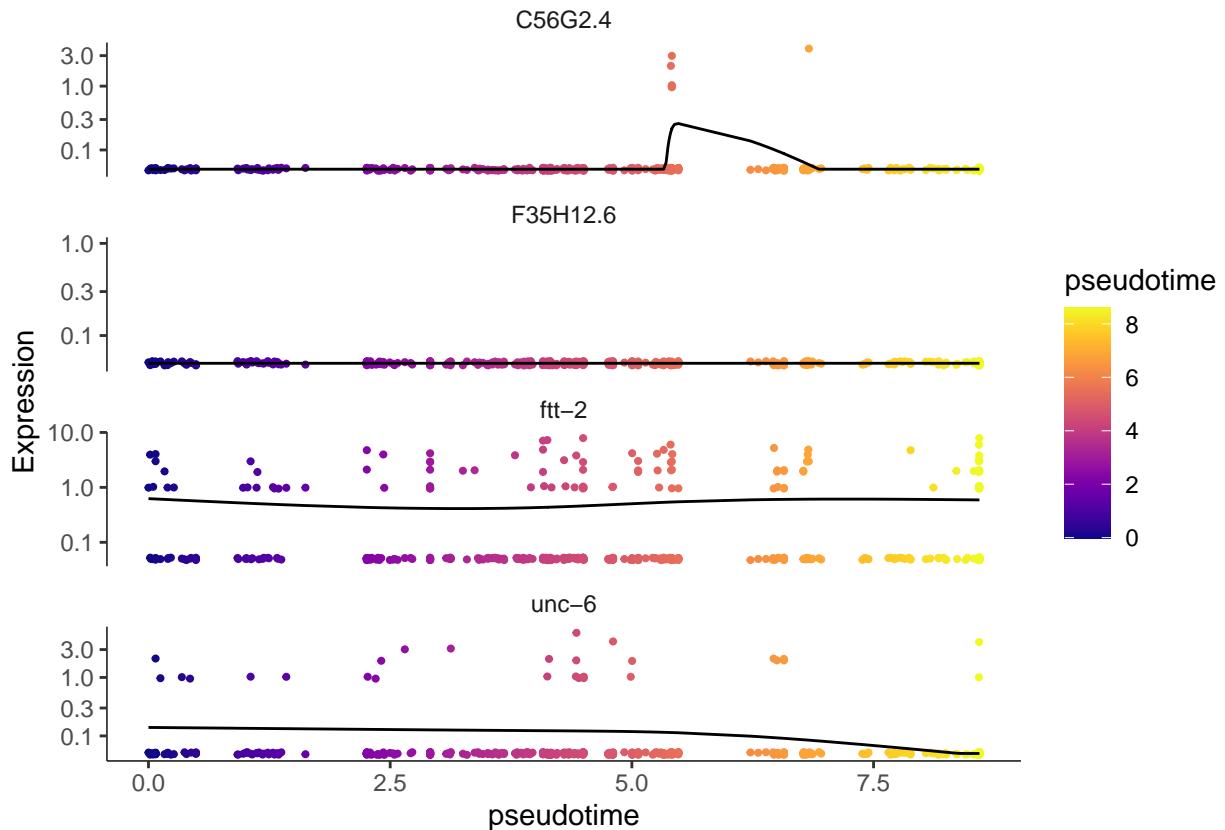
Differentially expressed dividing by 2 branches (cds_sub2.0)

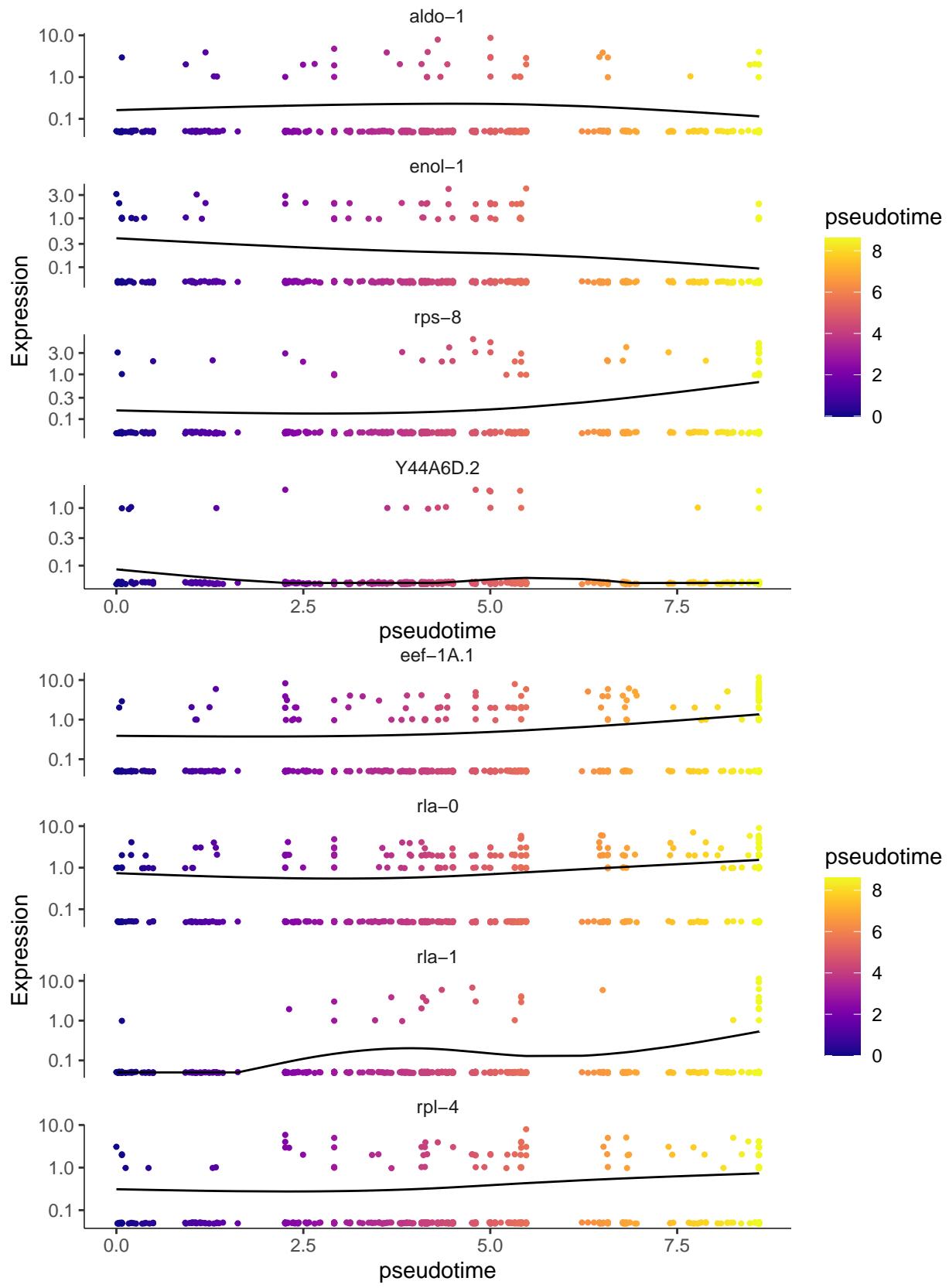
```

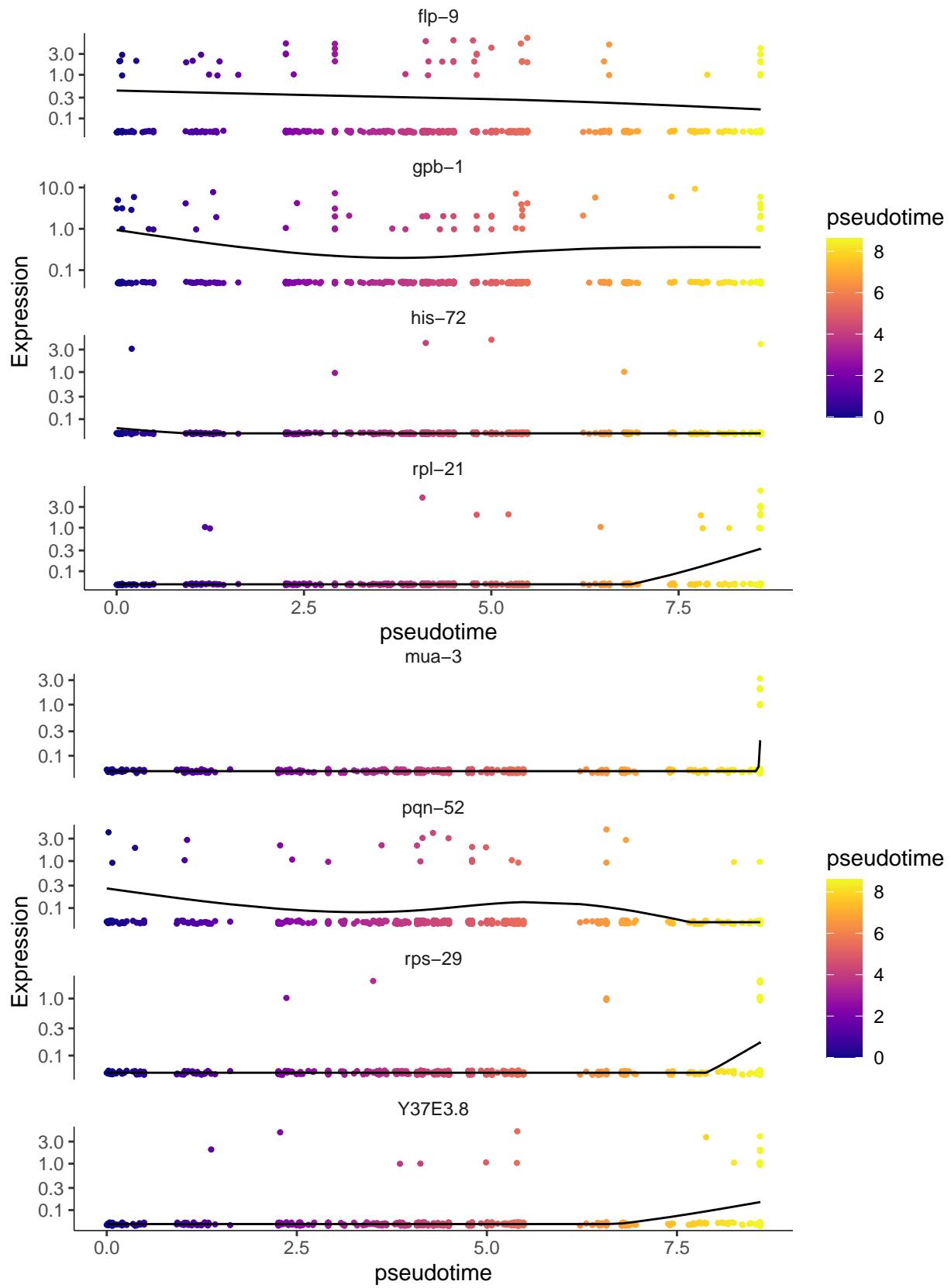
nplots <- 4
x <- seq_along(as.character(cds_genes$gene_short_name[1:40]))
toplot <- split(as.character(cds_genes$gene_short_name[1:40]), ceiling(x/nplots))
fillplot <- nplots-length(toplot[[length(toplot)]]])
if (fillplot != 0) {
  toplot[[length(toplot)]] <- c(toplot[[length(toplot)]], toplot[[1]][1:fillplot])
}

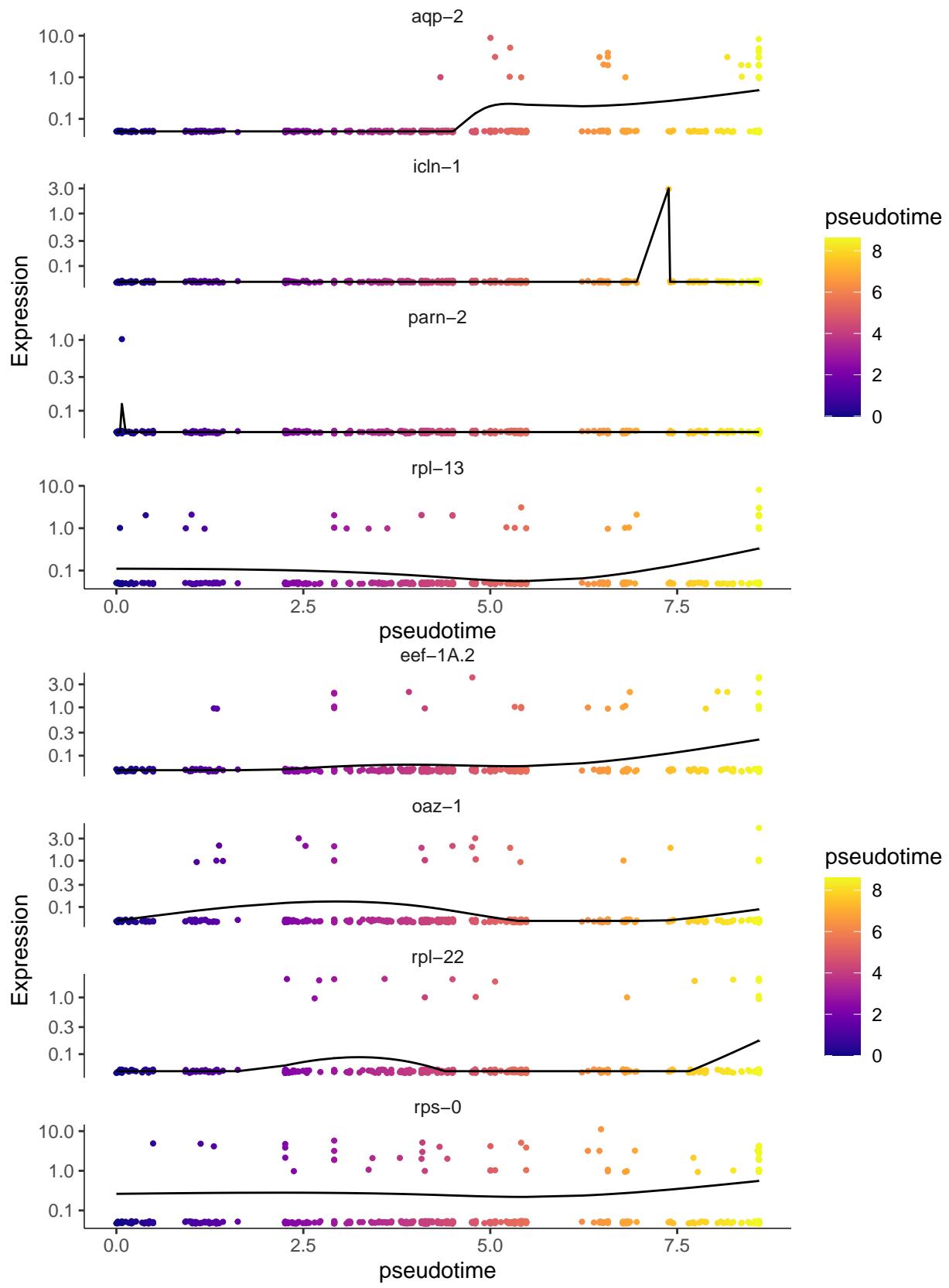
for (x in toplot) {
  cds_lineage_cds <- cds_sub2.0[rowData(cds_sub2.0)$gene_short_name %in% x,]
  print(plot_genes_in_pseudotime(cds_lineage_cds,
                                  min_expr=0.05))
}

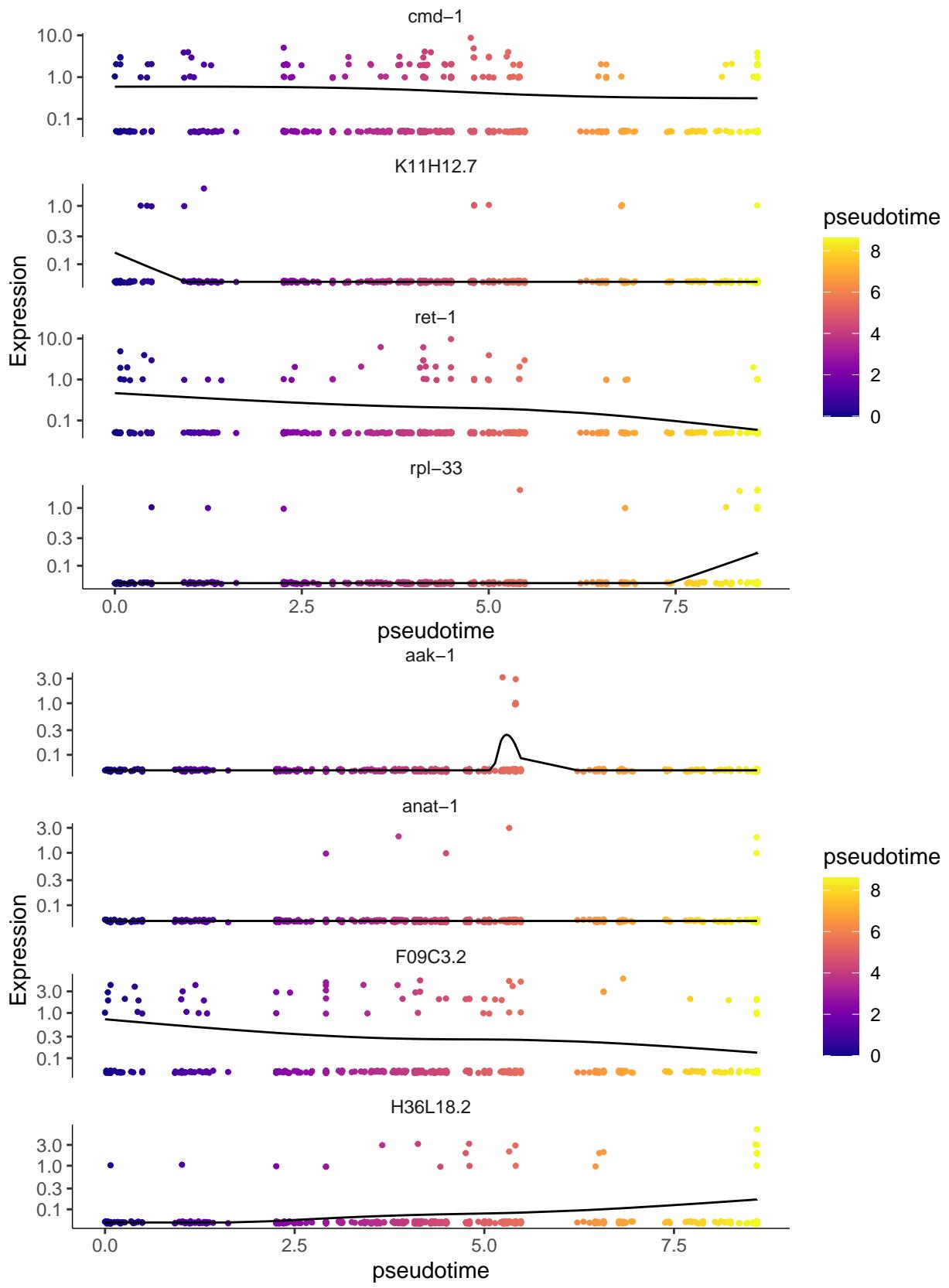
```

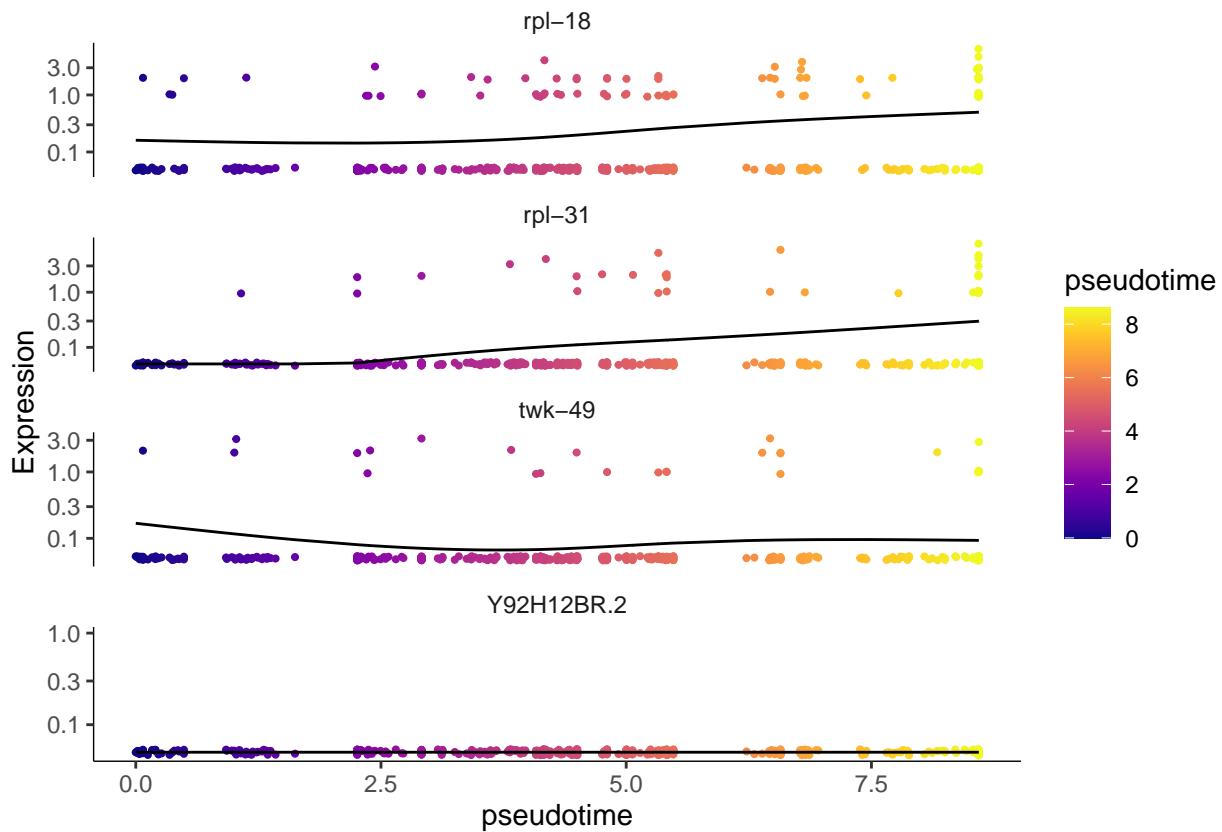








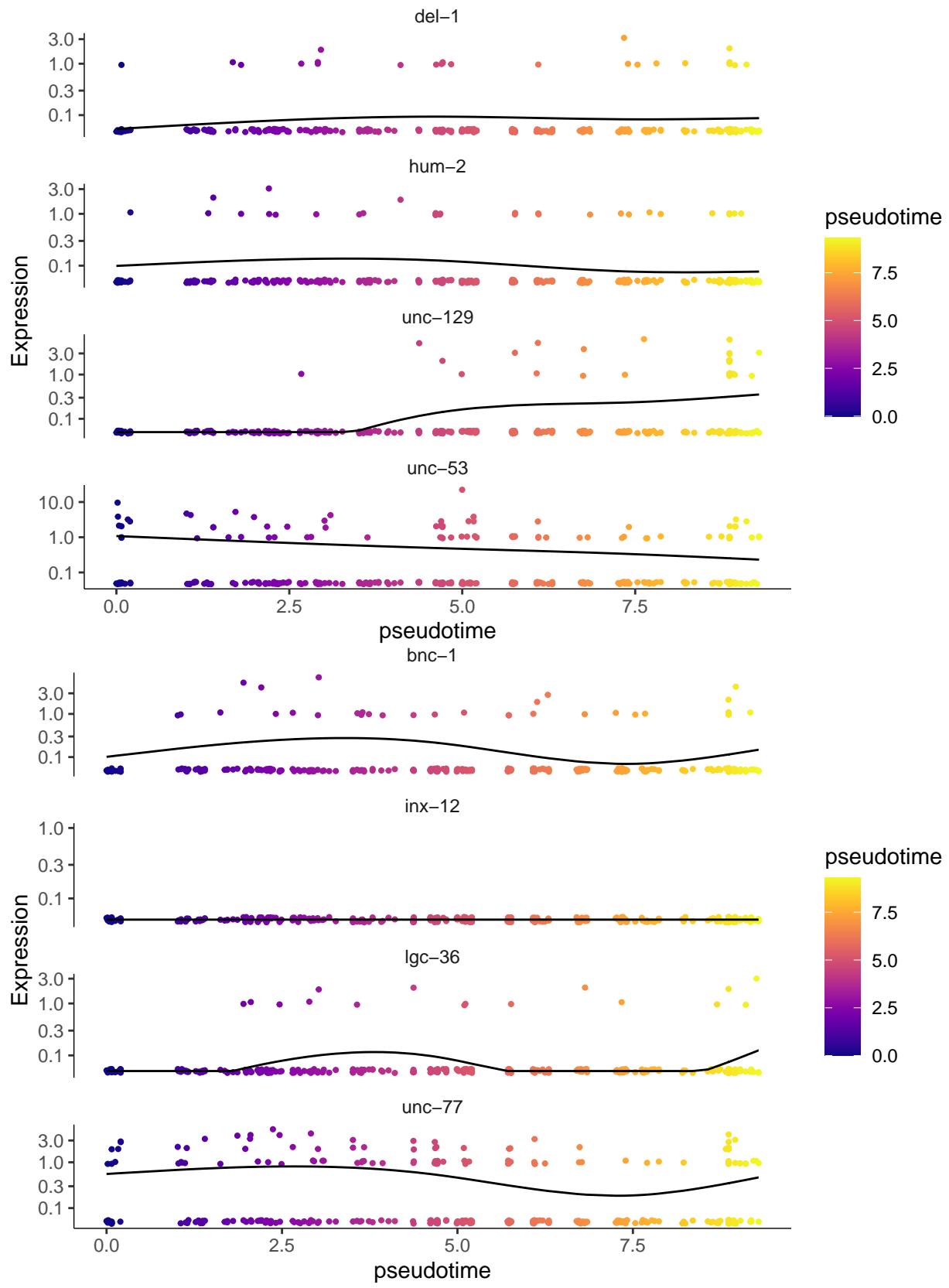


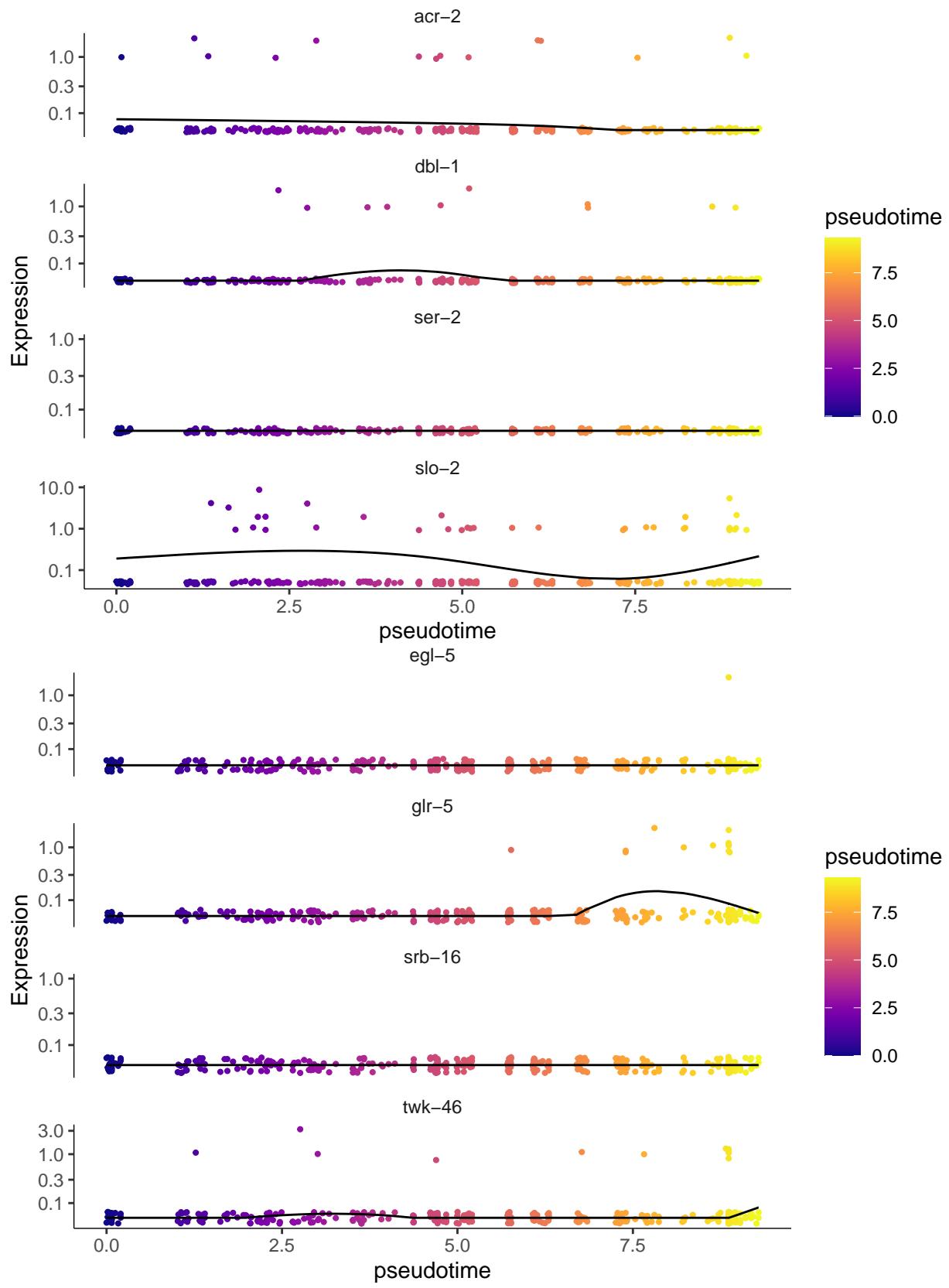


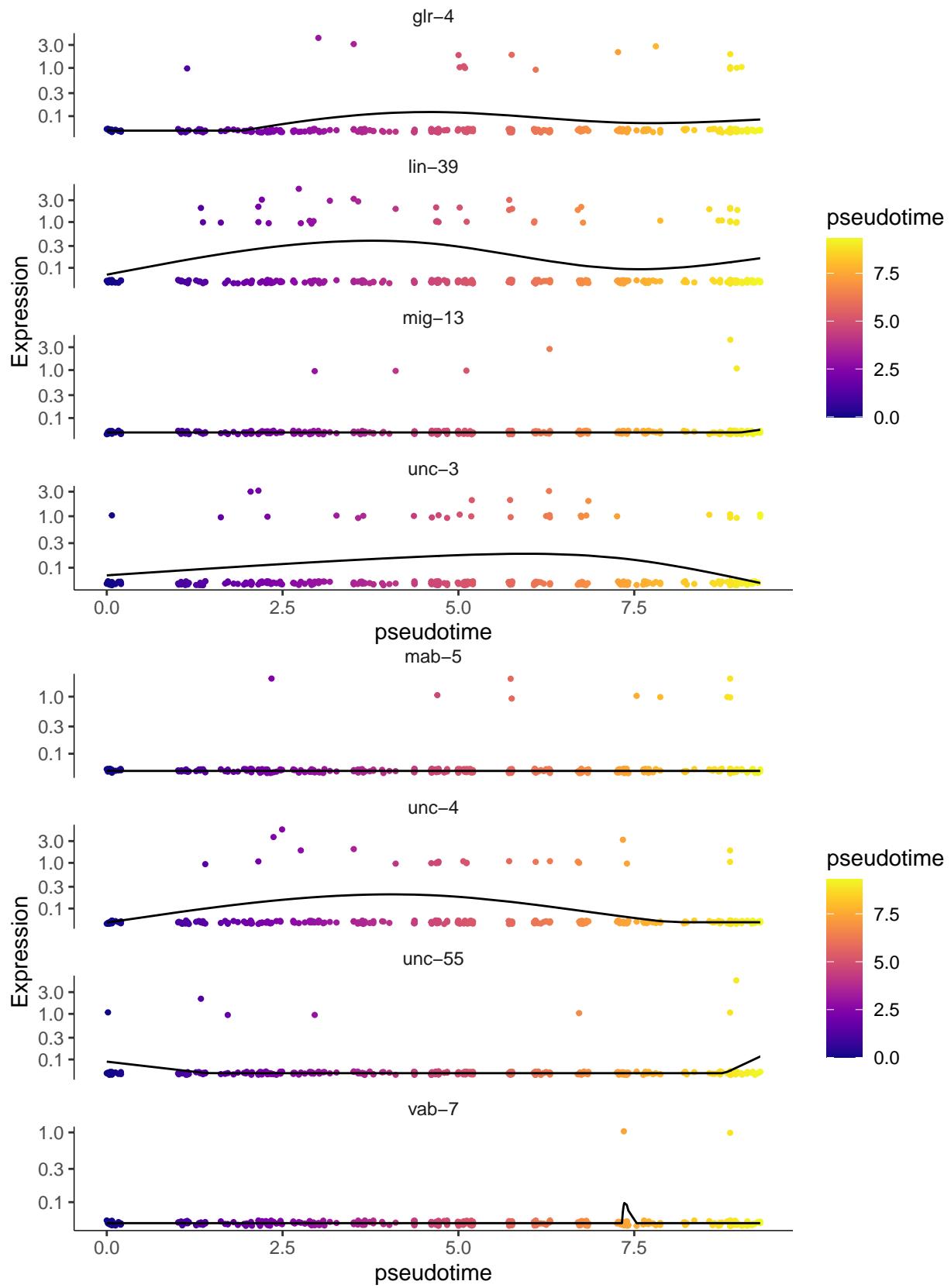
Selected genes cds_sub0.1

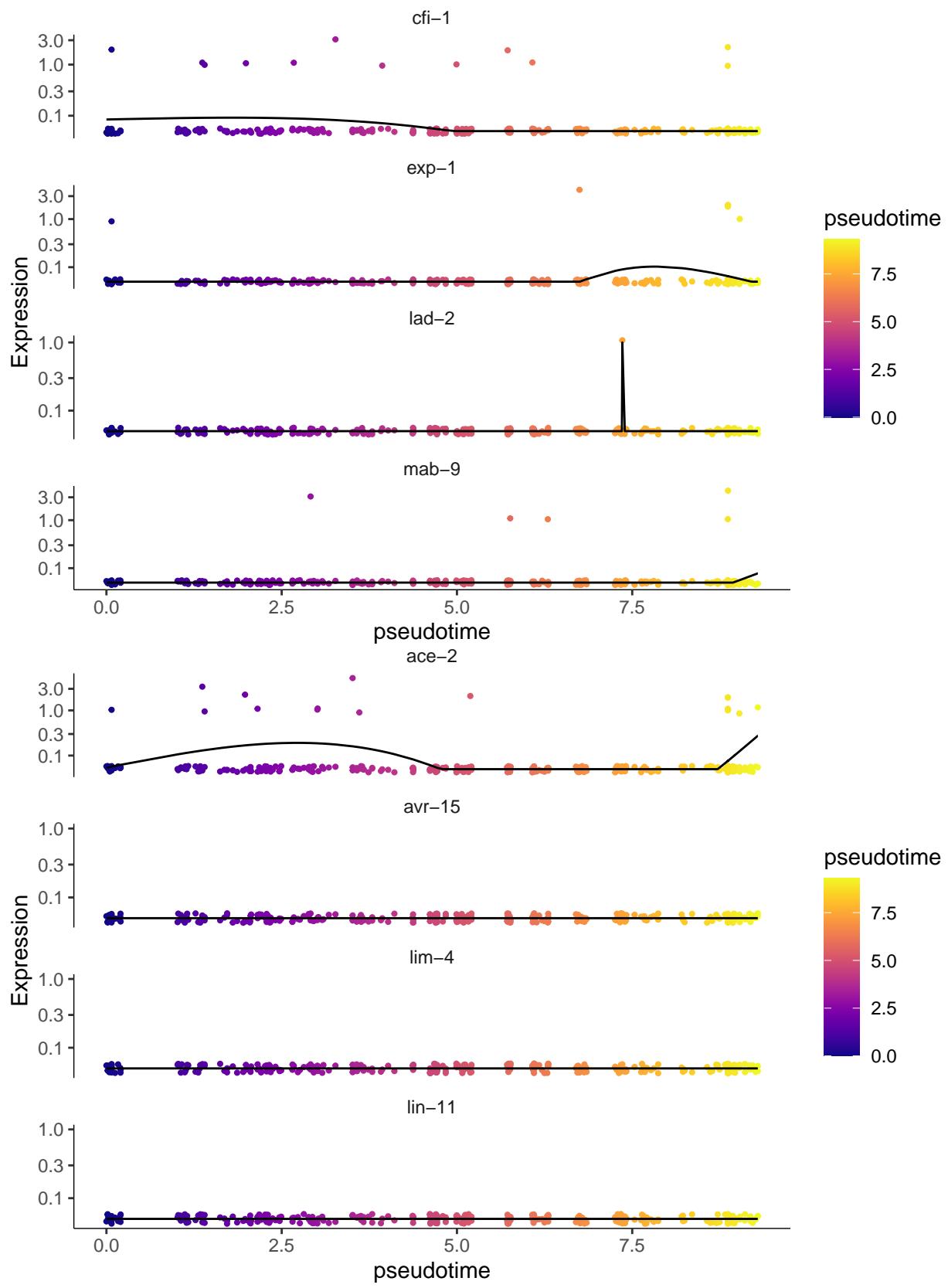
```
nplots <- 4
x <- seq_along(list_genes)
toplot <- split(list_genes, ceiling(x/nplots))
fillplot <- nplots-length(toplot)[[length(toplot)]]
toplot[[length(toplot)]] <- c(toplot[[length(toplot)]], toplot[[1]][1:fillplot])

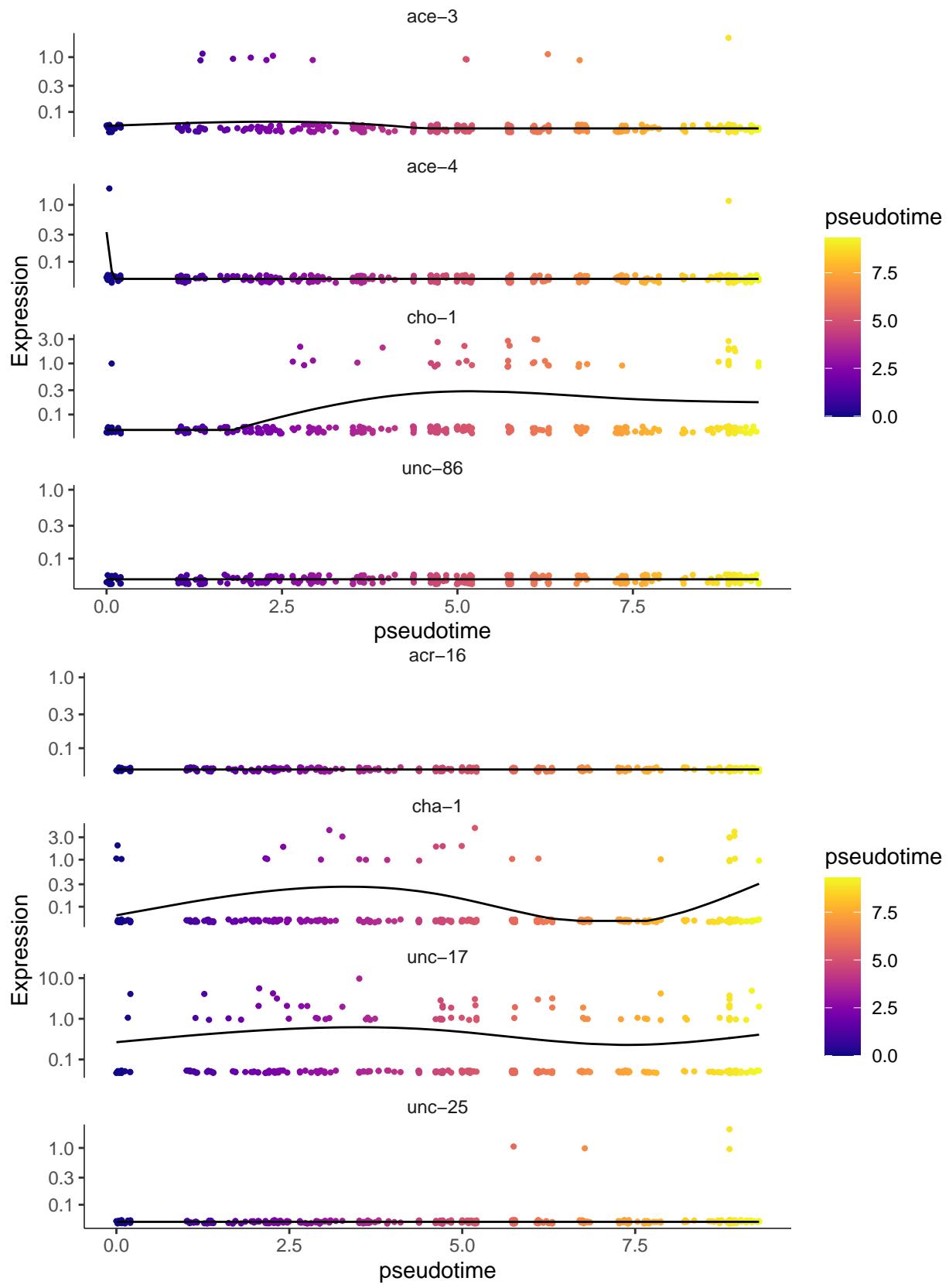
for (x in toplot) {
  cds_lineage_cds <- cds_sub0.1[rowData(cds_sub0.1)$gene_short_name %in% x,]
  print(plot_genes_in_pseudotime(cds_lineage_cds,
                                  min_expr=0.05))
}
```

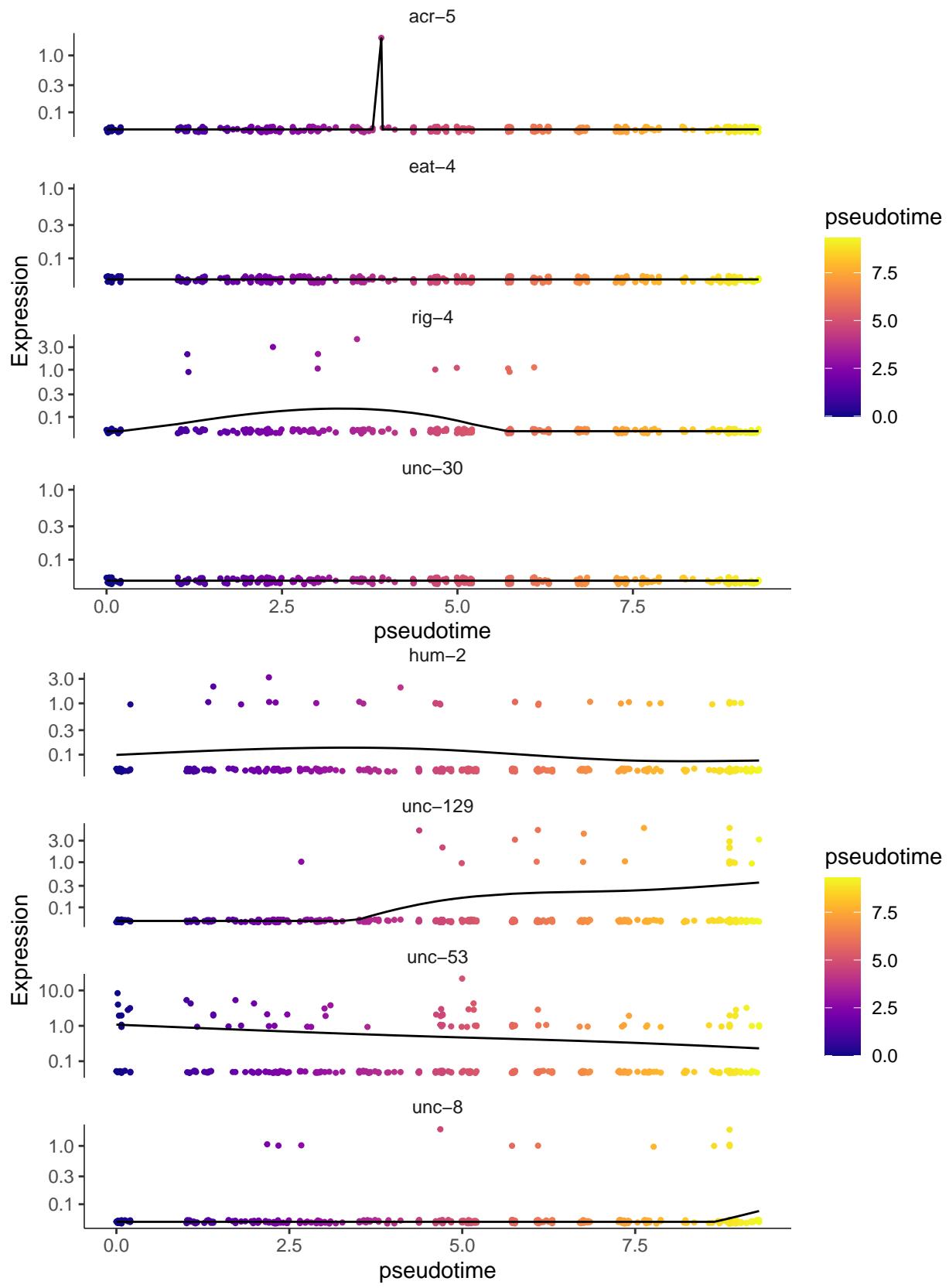












Selected genes cds_sub2.0

```

nplots <- 4
x <- seq_along(list_genes)
toplot <- split(list_genes, ceiling(x/nplots))
fillplot <- nplots-length(toplot)[length(toplot)]]
toplot[[length(toplot)]] <- c(toplot[[length(toplot)]], toplot[[1]][1:fillplot])
}

for (x in toplot) {
  cds_lineage_cds <- cds_sub2.0[rowData(cds_sub2.0)$gene_short_name %in% x,]
  print(plot_genes_in_pseudotime(cds_lineage_cds,
                                  min_expr=0.05))
}

```

