

human_pbmc_seurat

April 13, 2022

1 install iterbi

```
[1]: devtools::install_github("leezx/iterbi", dependencies = F)
```

Using github PAT from envvar GITHUB_PAT

Downloading GitHub repo leezx/iterbi@HEAD

```
checking for file '/private/var/folders/fp/bzk0_6352kl60dq4fjw
nx2z80000gn/T/RtmpIilJqX/remotes1060c2358ab75/leezx-
iterbi-100d0ec/DESCRIPTION' (377ms)
  preparing 'iterbi':
    checking DESCRIPTION meta-information
    checking for LF line-endings in source and make
files and shell scripts
    checking for empty or unneeded
directories
    building 'iterbi_0.4.9.tar.gz'
```

1.1 load packages

```
[1]: library(iterbi)
library(Seurat)
```

Bioconductor version '3.10' is out-of-date; the current release version '3.14' is available with R version '4.1'; see <https://bioconductor.org/install>

Registered S3 method overwritten by 'enrichplot':

```
method          from
fortify.enrichResult DOSE
```

2 download seurat dataset

```
[3]: # download Seurat test data
GetTestData()
```

3 Seurat standard analysis

```
[4]: # Load the PBMC dataset
seuset <- Read10X(data.dir = "filtered_gene_bc_matrices/hg19/")
# Initialize the Seurat object with the raw (non-normalized data).
seuset <- CreateSeuratObject(counts = seuset, project = "pbmc3k", min.cells = 3,
  ↪ min.features = 200)
seuset
```

Warning message:

"Feature names cannot have underscores ('_'), replacing with dashes ('-')"

An object of class Seurat

13714 features across 2700 samples within 1 assay

Active assay: RNA (13714 features)

```
[5]: seuset[["percent.mt"]] <- PercentageFeatureSet(seuset, pattern = "^MT-")
```

```
[6]: seuset <- subset(seuset, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 &
  ↪ percent.mt < 5)
```

```
[7]: seuset <- NormalizeData(seuset)
seuset <- FindVariableFeatures(seuset, selection.method = "vst", nfeatures = 2000)
seuset <- ScaleData(seuset, features = rownames(seuset))
```

Centering and scaling data matrix

```
[8]: seuset <- RunPCA(seuset, features = VariableFeatures(object = seuset), verbose = F)
```

```
[9]: # # error from latest Matrix version
# # Error in validObject(.Object): invalid class "Graph" object: superclass
  ↪ "Mnumeric" not defined in the
# # environment of the object's class
# packageVersion("Matrix")
```

```
# remove.packages("Matrix")
# packageurl <- "https://cran.r-project.org/src/contrib/Archive/Matrix/Matrix_1.
  ↪ 3-2.tar.gz"
# install.packages(packageurl, repos=NULL, type="source")
# packageVersion("Matrix")
# restart R kernel
```

```
[10]: seuset <- FindNeighbors(seuset, dims = 1:10)
```

Computing nearest neighbor graph

Computing SNN

```
[11]: seuset <- FindClusters(seuset, resolution = 0.5, verbose = F)
seuset <- RunUMAP(seuset, dims = 1:10, verbose = F)
```

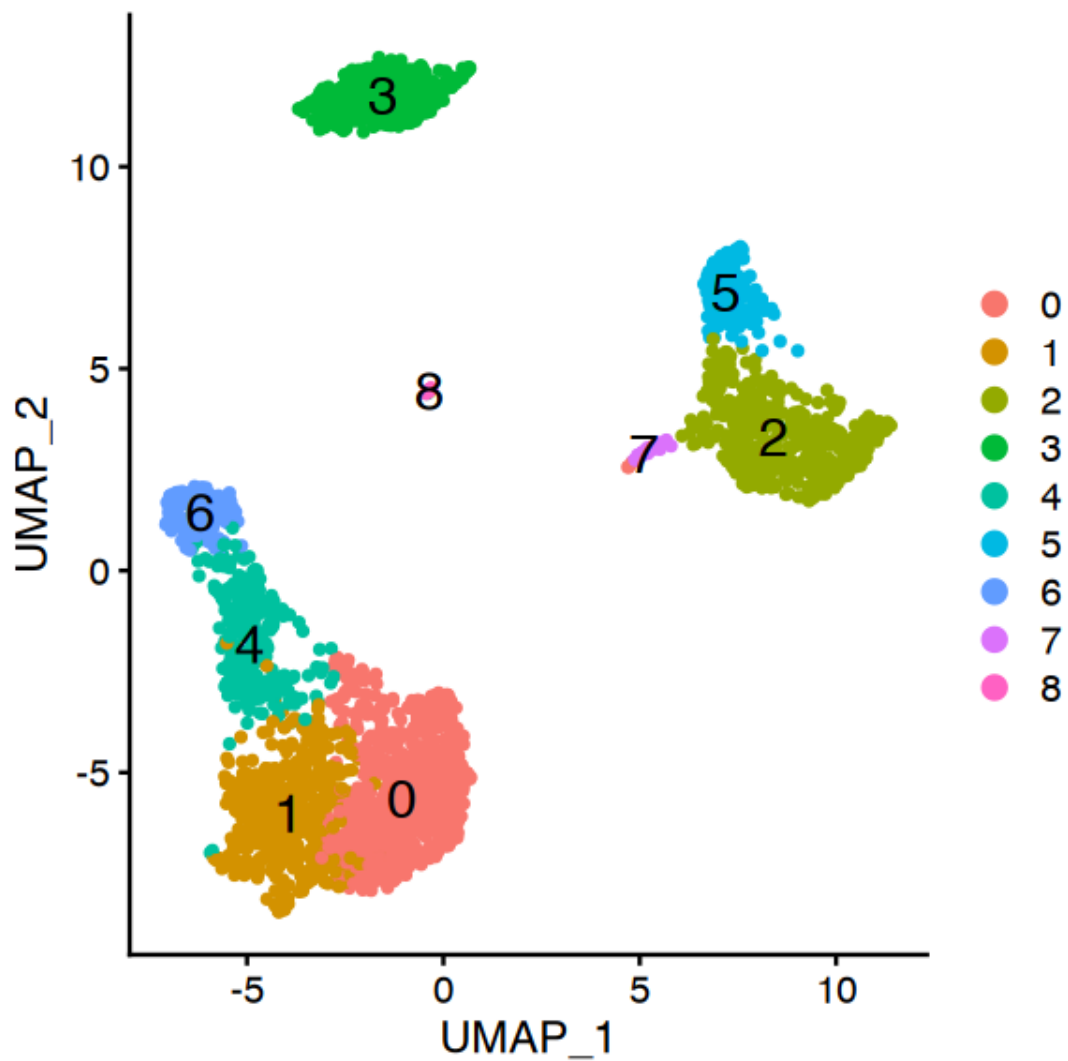
Warning message:

"The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R-native UWOT using the cosine metric
To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
This message will be shown once per session"

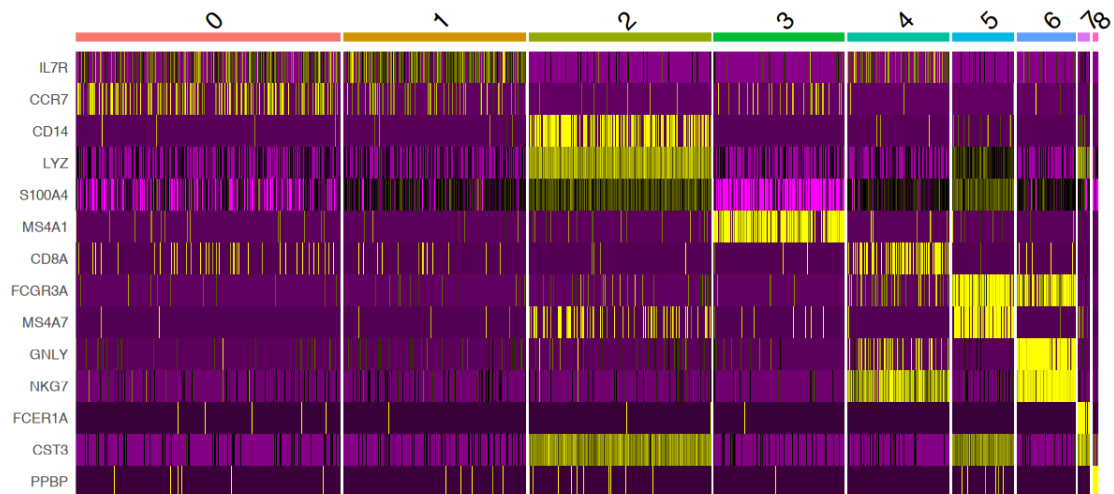
```
[12]: options(repr.plot.width=5, repr.plot.height=5)
DimPlot(seuset, reduction = "umap", label = TRUE, pt.size = 1, label.size = 6)
```

Warning message:

"Using `as.character()` on a quosure is deprecated as of rlang 0.3.0.
Please use `as_label()` or `as_name()` instead.
This warning is displayed once per session."



```
[13]: # show known markers
options(repr.plot.width=10, repr.plot.height=5)
DoHeatmap(seuset, features = c(
  ↪ c("IL7R", "CCR7", "CD14", "LYZ", "IL7R", "S100A4", "MS4A1", "CD8A",
      ↪ "FCGR3A", "MS4A7", "GNLY", "NKG7", "FCER1A", "CST3", "PPBP")) + NoLegend()
```

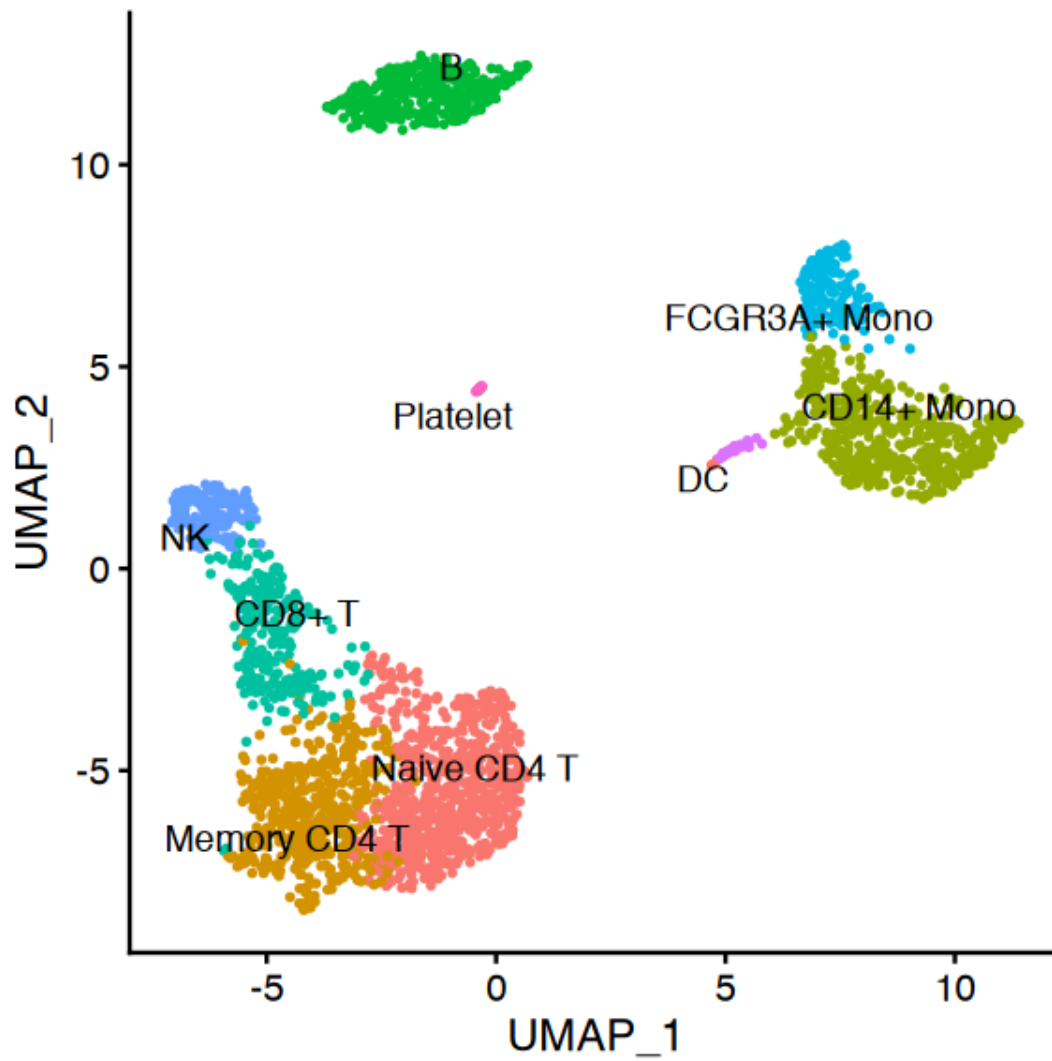


```
[14]: # 0 0 "Naive CD4+ T"
      # 1 2 "Memory CD4+"
      # 2 1 "CD14+ Mono"
      # 3 4 "B"
      # 4 3 "CD8+ T"
      # 5 5 "FCGR3A+ Mono"
      # 6 6 "NK"
      # 7 7 "DC"
      # 8 8 "Platelet"
```

```
[15]: new.cluster.ids <- c("Naive CD4 T", "Memory CD4 T", "CD14+ Mono", "B", "CD8+ T",
      "FCGR3A+ Mono",
      "NK", "DC", "Platelet")
```

```
[16]: names(new.cluster.ids) <- levels(seuset)
      seuset <- RenameIdents(seuset, new.cluster.ids)
```

```
[17]: options(repr.plot.width=5, repr.plot.height=5)
      DimPlot(seuset, reduction = "umap", label = TRUE, pt.size = 0.5, repel = T) +
      NoLegend()
```



```
[18]: # save the manual annotation
seuset$annotation <- seuset@active.ident
```

```
[19]: table(seuset$annotation)
```

Naive CD4 T	Memory CD4 T	CD14+ Mono	B	CD8+ T	FCGR3A+ Mono
697	483	480	344	271	162
NK	DC	Platelet			
155	32	14			

4 iterbi

- the index system in iterbi: L(x)_(y)
- L, level, the depth (x) of bifurcation, start from 0
- y is the cluster index at level x, start from 1

```
[2]: print(load("pbmc.seurat.Rdata"))
```

```
[1] "seuset"
```

4.1 iteratively bifurcated clustering

```
[3]: iterbi.result <- RunIterbi(seuset, min.marker.num = 30)
```

We are now at L0

Bifurcating L0_1 ...

Successfully split L0_1 to L1_1 and L1_2

We are now at L1

Bifurcating L1_1 ...

Successfully split L1_1 to L2_1 and L2_2

Bifurcating L1_2 ...

Successfully split L1_2 to L2_3 and L2_4

We are now at L2

Bifurcating L2_1 ...

Successfully split L2_1 to L3_1 and L3_2

Bifurcating L2_2 ...

Bifurcating L2_3 ...

Successfully split L2_3 to L3_5 and L3_6

Bifurcating L2_4 ...

only have 35 cells, set it as an end node

We are now at L3

```
Bifurcating L3_1 ...  
Successfully split L3_1 to L4_1 and L4_2  
  
Bifurcating L3_5 ...  
Successfully split L3_5 to L4_9 and L4_10  
  
Bifurcating L3_2 ...  
Successfully split L3_2 to L4_3 and L4_4  
  
Bifurcating L3_6 ...  
only have 13 cells, set it as an end node  
  
We are now at L4  
  
Bifurcating L4_1 ...  
  
Bifurcating L4_9 ...  
Successfully split L4_9 to L5_17 and L5_18  
  
Bifurcating L4_4 ...  
  
Bifurcating L4_3 ...  
  
Bifurcating L4_10 ...  
  
Bifurcating L4_2 ...  
  
We are now at L5  
  
Bifurcating L5_17 ...  
  
Bifurcating L5_18 ...  
  
We are now at L6  
  
Bifurcating stopped! No more clusters can be split
```

```
[4]: save(iterbi.result, file = "pbmc.raw.iterbi.result.Rdata")
```


4.2 order clusters

```
[10]: print(load("pbmc.raw.iterbi.result.Rdata"))
```

```
[1] "iterbi.result"
```

```
[5]: iterbi.result <- OrderCluster(seuset, iterbi.result)
```

```
switch 3 4 at level L2
```

```
switch 1 2 at level L3
```

4.3 rename clusters

```
[6]: iterbi.result <- RenameIterbi(iterbi.result)
```

The following `from` values were not present in `x`: L0_1, L2_3, L4_2, L5_18

The following `from` values were not present in `x`: L2_2, L2_4, L3_6, L4_3, L4_4, L4_1, L4_2, L4_10, L5_17, L5_18

The following `from` values were not present in `x`: L2_2, L2_4, L3_6, L4_3, L4_4, L4_1, L4_2, L4_10, L5_17, L5_18

The following `from` values were not present in `x`: L0_1, L1_2, L2_2, L2_4, L3_2, L3_6, L4_4, L4_2, L4_10, L5_18

The following `from` values were not present in `x`: L0_1, L1_1, L2_1, L2_3, L3_1, L3_5, L4_3, L4_1, L4_9, L5_17

4.4 marker details

classification of marker - expression level changed marker - have very high expression pct in both target and background - uniquely expressed marker in one cluster - uniquely expressed marker in multiple cluster

```
[7]: iterbi.result$marker_chain <- AddMarkerExpressionPct(seuset, iterbi.  
  ↪result$cellMeta, iterbi.result$marker_chain)
```

4.5 write iterbi result into seurat

```
[10]: seuset <- WriteIterbiIntoSeurat(seuset, iterbi.result)
```

4.6 GO annotation (optional)

```
[8]: # # remove duplicate markers, each marker only has one best matched cluster
# iterbi.marker.chain.rmDup <- RemoveDuplicatedMarker(iterbi.
  ↪result$marker_chain)
```

```
[11]: iterbi.GO.anno <- IterbiEnrichGO(iterbi.result$marker_chain, organism = "hs",
  ↪pvalueCutoff = 0.05, min_count = 3)
```

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

'select()' returned 1:many mapping between keys and columns

```
[12]: table(iterbi.GO.anno$cluster)
```

L1_1	L1_2	L2_1	L2_2	L2_3	L3_1	L3_2	L3_5	L3_6	L4_1	L4_2	L4_3	L4_7	L4_8	L5_8
10	22	14	6	4	15	4	4	4	4	11	13	17	8	4

```
[13]: seuset@assays$iterbi$GO_chain <- iterbi.GO.anno
```

4.7 save

```
[14]: save(seuset, file = "pbmc.seurat.Rdata")
```

4.8 visualization

```
[17]: print(load("pbmc.seurat.Rdata"))
```

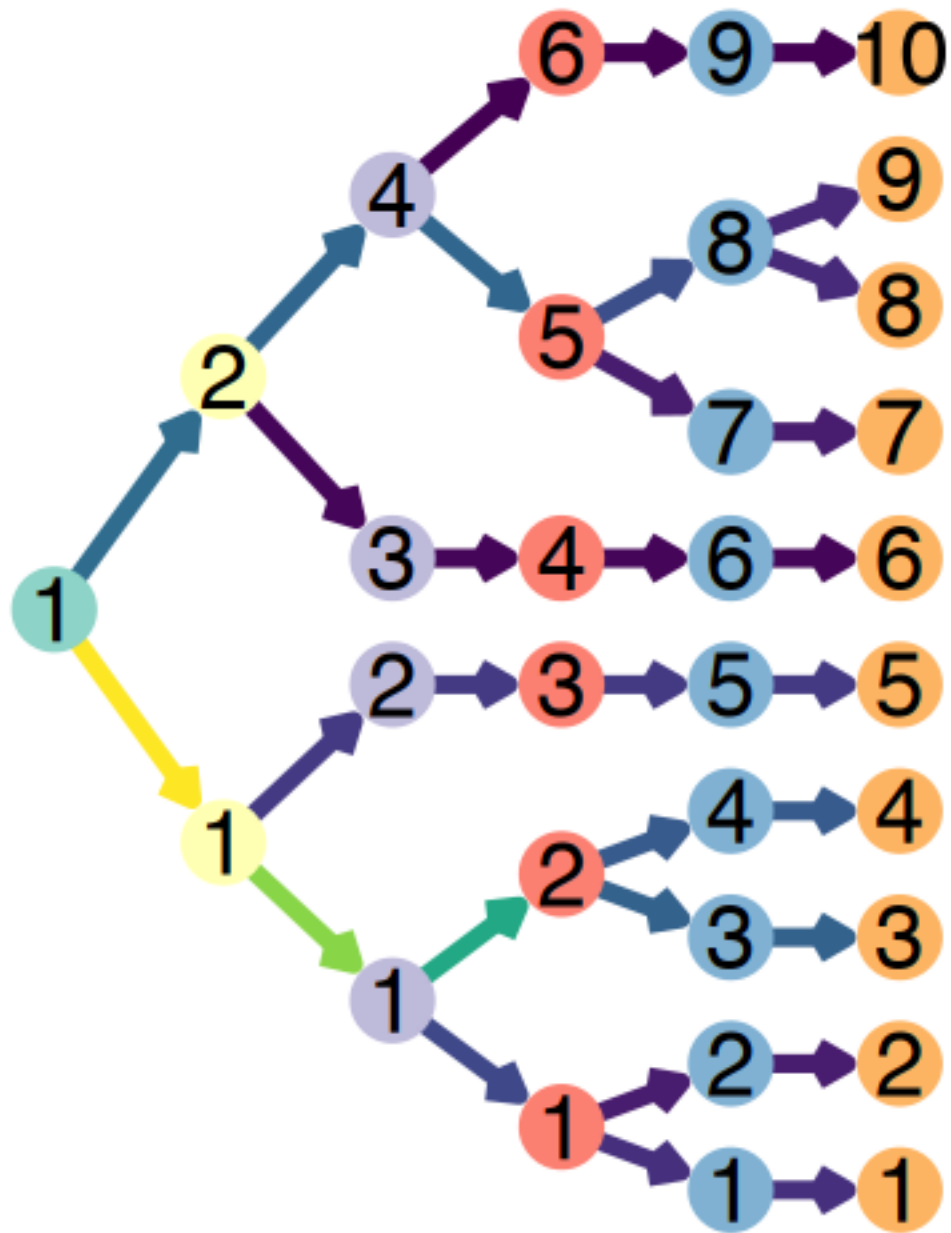
```
[1] "seuset"
```

```
[15]: iterbi.cellMeta <- seuset@assays$iterbi[["cellMeta"]]  
      iterbi.marker.chain <- seuset@assays$iterbi[["marker_chain"]]  
      iterbi.bifucation <- seuset@assays$iterbi[["bifucation"]]
```

```
[16]: go.marker.chain <- seuset@assays$iterbi[["GO_chain"]]
```

4.8.1 binary tree

```
[17]: options(repr.plot.width=3, repr.plot.height=4)  
      DrawIterbiClusterTree(seuset, iterbi.cellMeta)
```

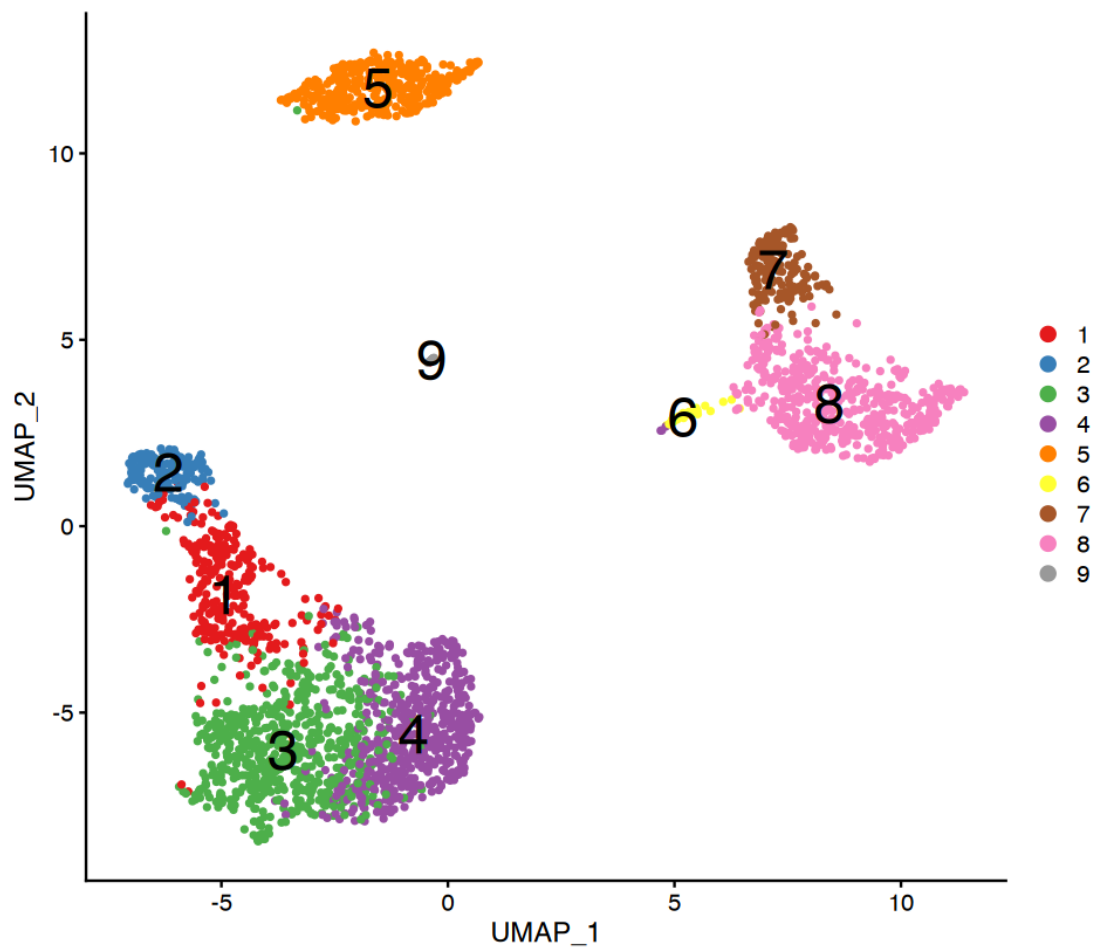


4.8.2 UMAP or tSNE

```
[18]: library(ggplot2)
```

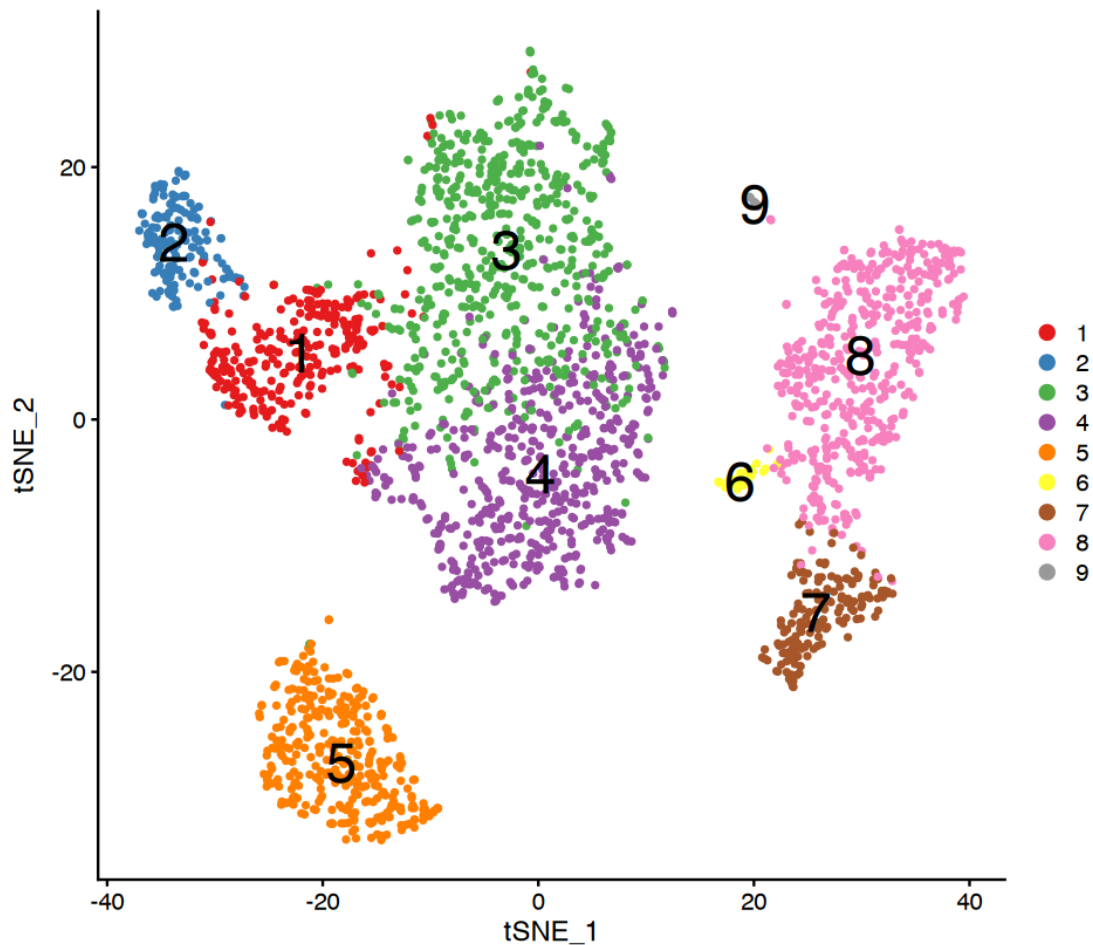
```
[19]: # similar group have continous number
options(repr.plot.width=8, repr.plot.height=7)
DimPlot(seuset, reduction = "umap", group.by = "L4", pt.size = 1, label = T,
  ↪label.size = 10) +
```

```
scale_color_manual(values=IterbiColors())
```



```
[20]: seuset <- RunTSNE(seuset, dims = 1:10)
```

```
[21]: options(repr.plot.width=8, repr.plot.height=7)
DimPlot(seuset, reduction = "tsne", group.by = "L4", pt.size = 1, label = T,
  ↪label.size = 10) +
  scale_color_manual(values=IterbiColors())
```



4.8.3 heatmap

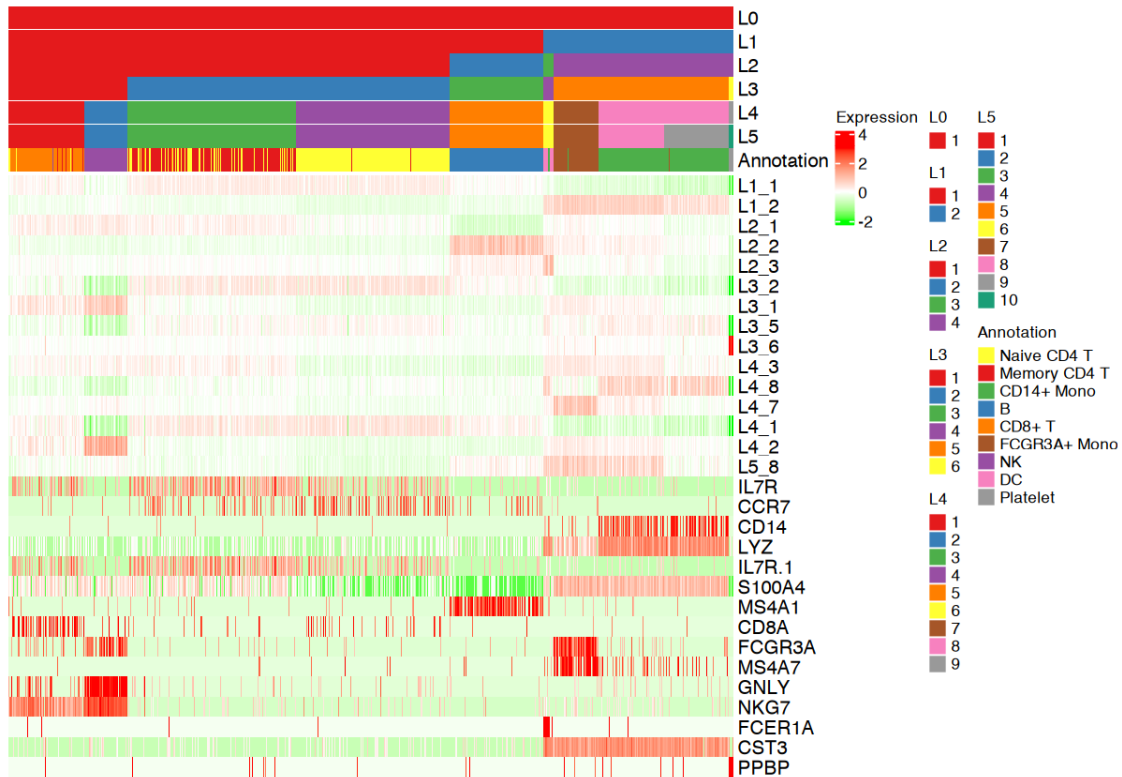
```
[22]: known_markers <- c(
  ↪c("IL7R", "CCR7", "CD14", "LYZ", "IL7R", "S100A4", "MS4A1", "CD8A", "FCGR3A", "MS4A7", "GNLY", "NKG7",
    ↪"FCER1A", "CST3", "PPBP")
```

```
[39]: iterbi.cellMeta.factor <- iterbi.cellMeta
for (i in colnames(iterbi.cellMeta.factor)) {
  iterbi.cellMeta.factor[,i] <- factor(iterbi.cellMeta.factor[,i], levels = c(
  ↪unique(iterbi.cellMeta.factor[,i]))
}
```

```
[55]: # levels(iterbi.cellMeta.factor$L5)
```

```
[45]: options(repr.plot.width=10, repr.plot.height=7)
```

```
DrawMarkerChainHeatmap(seuset, iterbi.cellMeta.factor, iterbi.marker.chain,
  ↪compare_anno="annotation",
  known_markers=known_markers)
```



4.8.4 dotplot

```
[24]: library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
options(repr.plot.width=20, repr.plot.height=7)
DrawMarkerChainDotplot(seuset, iterbi.marker.chain)
```



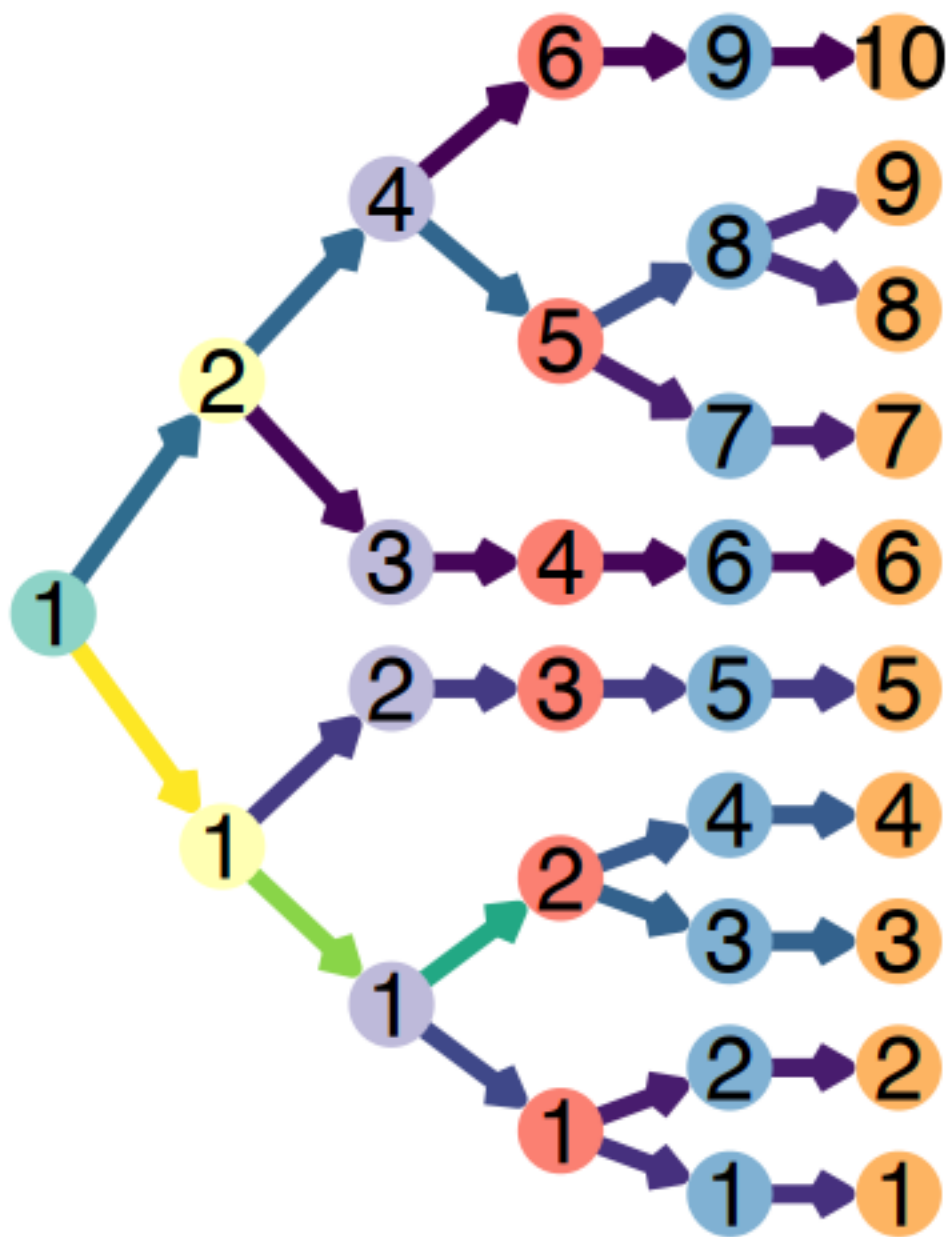
```
# filter markers
iterbi.marker.chain.uniq <- subset(iterbi.marker.chain, (cluster_pct-bcg_pct)>0.
  ↪ 3 & cluster_pct>0.7 & bcg_pct<0.5)
```

```
options(repr.plot.width=15, repr.plot.height=7)
DrawMarkerChainDotplot(seuset, iterbi.marker.chain.uniq)
```



4.9 module chain

```
options(repr.plot.width=3, repr.plot.height=4)
DrawIterbiClusterTree(seuset, iterbi.cellMeta)
```

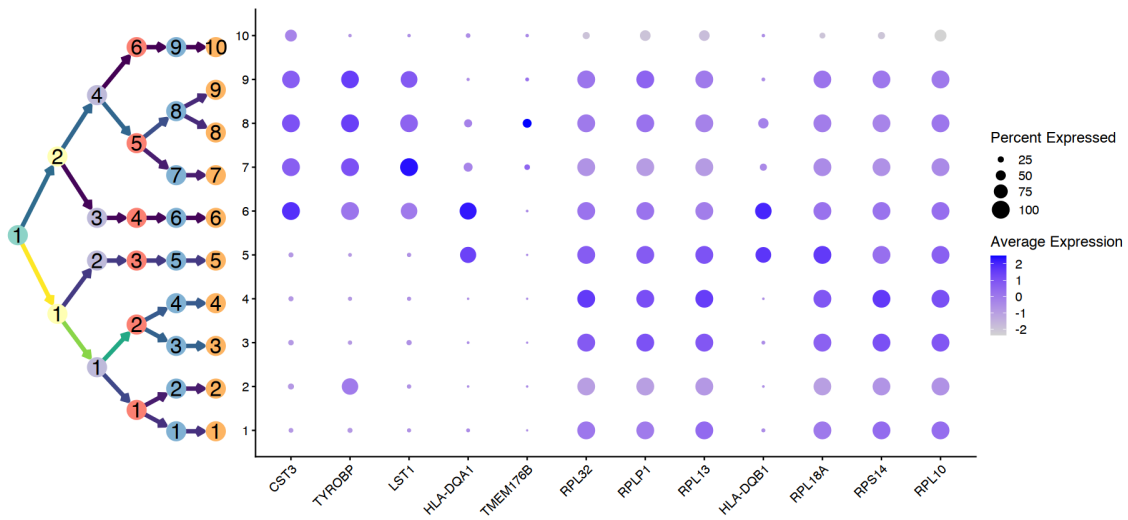



```
[46]: cluster.chain <- GetClusterChain(iterbi.cellMeta, iterbi.bifucation, "L5_8")
      cluster.chain
```

```
L0\_1 'Parent_node' L1\_2 'Parent_node' L2\_4 'Parent_node' L3\_5 'Parent_node'
L4\_8 'Parent_node' L5\_8 'Parent_node'
```

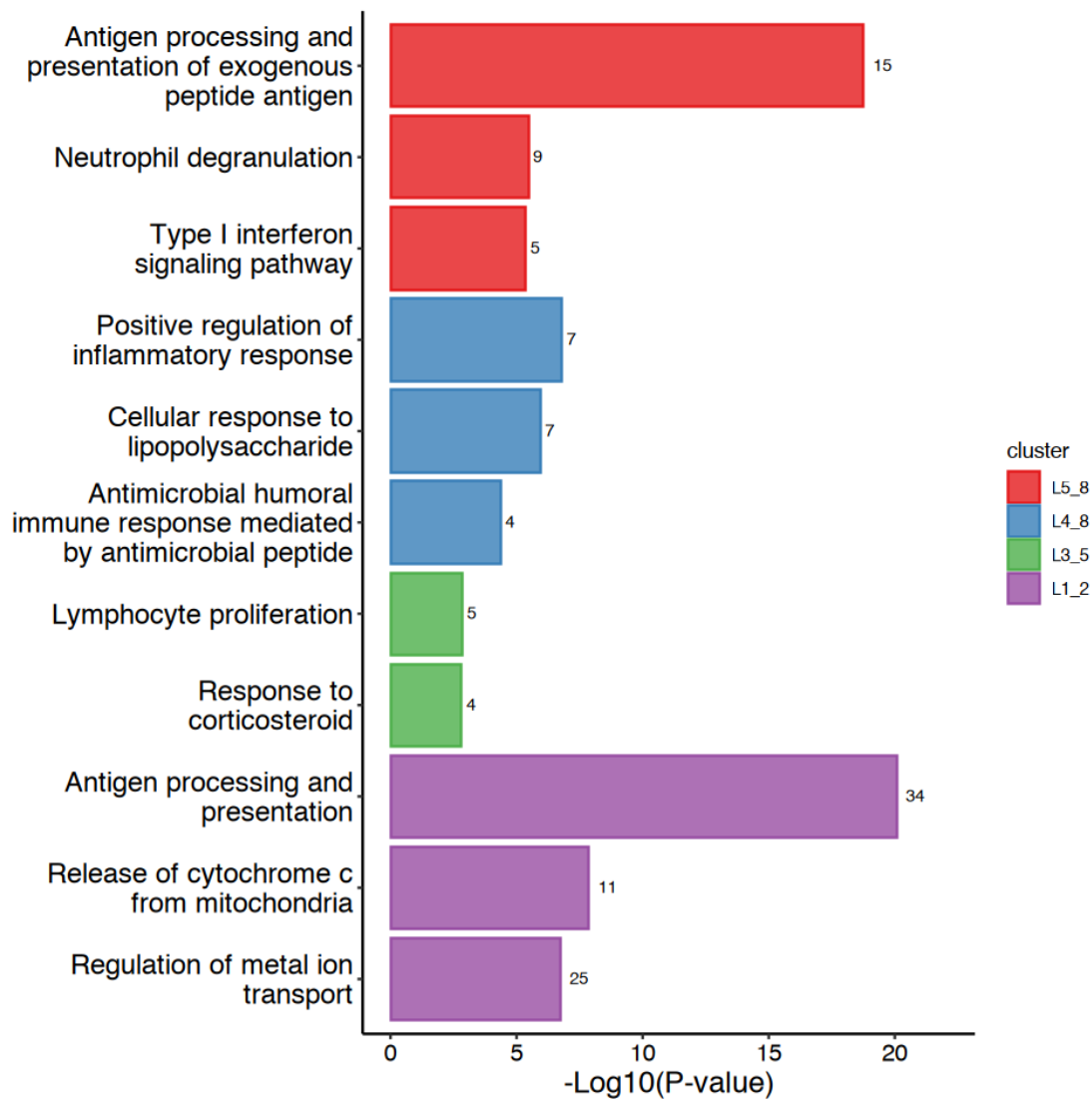
```
[47]: target.marker.chain <- subset(iterbi.marker.chain, cluster %in% names(cluster.
  ↪chain))
```

```
[48]: options(repr.plot.width=15, repr.plot.height=7)
      DrawMarkerChainDotplot(seuset, target.marker.chain)
```



4.10 GO chain

```
[49]: options(repr.plot.width=8, repr.plot.height=8)
      DrawGOchain(seuset@assays$iterbi$GO_chain, cluster.chain)
```



4.11 initial bifurcation event

```
[50]: # different level, different cluster
cluster_1 <- "L5_8"
cluster_2 <- "L5_9"
```

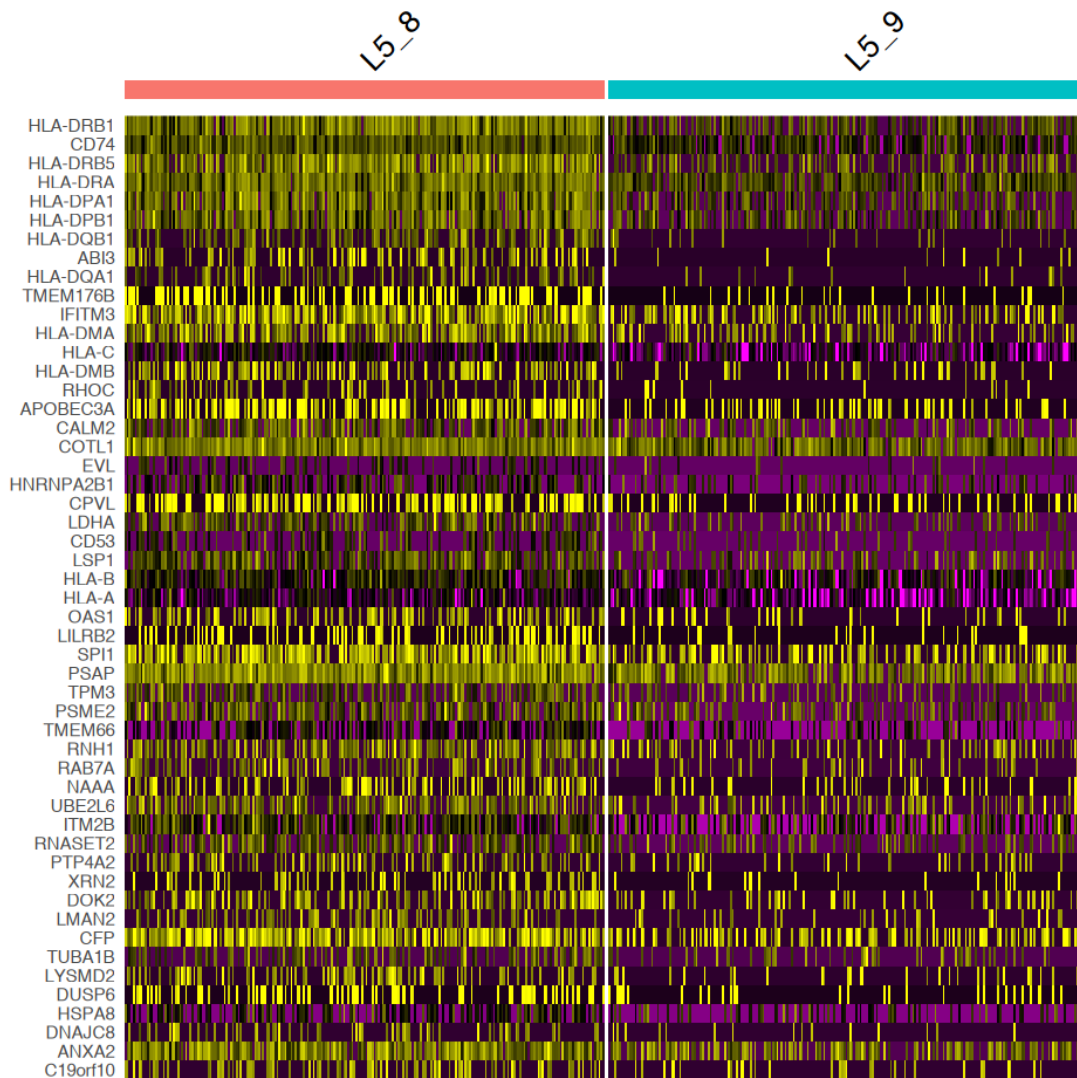
```
[ ]: # different level, same cluster
cluster_1 <- "L3_1"
cluster_2 <- "L4_1"
```

```
[42]: # same level, different cluster
cluster_1 <- "L4_1"
cluster_2 <- "L4_2"
```

```
[51]: iterbi.tracer <- GetInitialBifurcation(seuset, iterbi.cellMeta, cluster_1,
      ↪cluster_2)

[52]: tmp.seuset <- iterbi.tracer[["subset_seuratObj"]]
      diff_marker_1 <- iterbi.tracer[["diff_marker_1"]]
      diff_marker_2 <- iterbi.tracer[["diff_marker_2"]]

[53]: options(repr.plot.width=8, repr.plot.height=8)
      DoHeatmap(tmp.seuset, features = rbind(diff_marker_1, diff_marker_2)$gene) +
      NoLegend()
```



5 sessionInfo

```
[54]: sessionInfo()
```

```
R version 3.6.2 (2019-12-12)
```

```
Platform: x86_64-apple-darwin15.6.0 (64-bit)
```

```
Running under: macOS Mojave 10.14.6
```

```
Matrix products: default
```

```
BLAS: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
```

```
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
```

```
locale:
```

```
[1] C/UTF-8/C/C/C/C
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] dplyr_1.0.8  ggplot2_3.3.5 Seurat_3.1.4  iterbi_0.4.9
```

```
loaded via a namespace (and not attached):
```

[1] utf8_1.2.1	reticulate_1.24	tidyselect_1.1.1
[4] RSQLite_2.2.0	AnnotationDbi_1.48.0	htmlwidgets_1.5.1
[7] BiocParallel_1.20.1	grid_3.6.2	Rtsne_0.15
[10] munsell_0.5.0	codetools_0.2-16	mutoss_0.1-12
[13] ica_1.0-2	pbdZMQ_0.3-3	future_1.24.0
[16] withr_2.5.0	GOSeqSim_2.12.0	colorspace_2.0-2
[19] Biobase_2.46.0	knitr_1.38	uuid_0.1-4
[22] rstudioapi_0.13	stats4_3.6.2	ROCR_1.0-7
[25] DOSE_3.12.0	gbRd_0.4-11	listenv_0.8.0
[28] labeling_0.4.2	Rdpack_0.11-1	urltools_1.7.3
[31] repr_1.1.0	mnormt_1.5-6	polyclip_1.10-0
[34] bit64_4.0.5	farver_2.1.0	rprojroot_2.0.3
[37] parallelly_1.31.0	vctrs_0.3.8	generics_0.1.2
[40] TH.data_1.0-10	xfun_0.29	R6_2.5.0
[43] doParallel_1.0.16	clue_0.3-60	graphlayouts_0.7.1
[46] rsvd_1.0.3	gridGraphics_0.5-1	fgsea_1.12.0
[49] bitops_1.0-6	cachem_1.0.6	assertthat_0.2.1
[52] scales_1.1.1	enrichplot_1.6.1	multcomp_1.4-12
[55] ggraph_2.0.5	nnet_7.3-12	gtable_0.3.0
[58] npsurv_0.4-0	org.Mm.eg.db_3.10.0	Cairo_1.5-12.2
[61] globals_0.14.0	tidygraph_1.2.0	sandwich_2.5-1
[64] rlang_1.0.1	GlobalOptions_0.1.2	splines_3.6.2
[67] lazyeval_0.2.2	europepmc_0.3	checkmate_2.0.0
[70] BiocManager_1.30.12	reshape2_1.4.4	backports_1.1.9
[73] qvalue_2.18.0	Hmisc_4.4-1	clusterProfiler_3.14.3
[76] tools_3.6.2	ggplotify_0.0.5	ellipsis_0.3.2

[79]	gplots_3.0.3	RColorBrewer_1.1-2	BiocGenerics_0.32.0
[82]	clustree_0.4.3	ggribges_0.5.2	TFisher_0.2.0
[85]	Rcpp_1.0.8.3	plyr_1.8.6	progress_1.2.2
[88]	base64enc_0.1-3	purrr_0.3.4	prettyunits_1.1.1
[91]	rpart_4.1-15	pbapply_1.4-3	GetoptLong_1.0.5
[94]	viridis_0.6.2	cowplot_1.0.0	S4Vectors_0.24.4
[97]	zoo_1.8-7	ggrepel_0.9.1	cluster_2.1.0
[100]	here_1.0.1	magrittr_2.0.2	magick_2.7.1
[103]	data.table_1.14.2	DO.db_2.9	circlize_0.4.14
[106]	triebeard_0.3.0	lmtest_0.9-39	RANN_2.6.1
[109]	mvtnorm_1.1-1	parallelDist_0.2.6	fitdistrplus_1.0-14
[112]	matrixStats_0.61.0	hms_1.1.1	patchwork_1.0.1.9000
[115]	lsei_1.2-0	evaluate_0.15	jpeg_0.1-8.1
[118]	IRanges_2.20.2	gridExtra_2.3	shape_1.4.6
[121]	compiler_3.6.2	tibble_3.1.6	KernSmooth_2.23-16
[124]	ncdf4_1.19	crayon_1.4.1	htmltools_0.5.2
[127]	Formula_1.2-3	tidyr_1.2.0	RcppParallel_5.0.2
[130]	DBI_1.1.2	tweenr_1.0.2	ComplexHeatmap_2.9.4
[133]	MASS_7.3-55	Matrix_1.3-2	cli_3.1.1
[136]	gdata_2.18.0	parallel_3.6.2	metap_1.3
[139]	igraph_1.2.11	pkgconfig_2.0.3	sn_1.5-5
[142]	rvcheck_0.1.8	numDeriv_2016.8-1.1	foreign_0.8-72
[145]	IRdisplay_0.7.0	plotly_4.9.2	xml2_1.3.2
[148]	foreach_1.5.1	multtest_2.42.0	bibtex_0.4.2.2
[151]	stringr_1.4.0	digest_0.6.28	sctransform_0.2.1
[154]	RcppAnnoy_0.0.15	tsne_0.1-3	fastmatch_1.1-0
[157]	leiden_0.3.3	htmlTable_2.0.1	uwot_0.1.5
[160]	gtools_3.8.2	rjson_0.2.20	lifecycle_1.0.1
[163]	nlme_3.1-142	jsonlite_1.7.2	HiClimR_2.2.1
[166]	viridisLite_0.4.0	fansi_0.4.2	pillar_1.7.0
[169]	lattice_0.20-45	GO.db_3.10.0	fastmap_1.1.0
[172]	httr_1.4.2	plotrix_3.7-8	survival_3.2-13
[175]	glue_1.6.1	png_0.1-7	iterators_1.0.13
[178]	bit_4.0.4	ggforce_0.3.3	stringi_1.5.3
[181]	blob_1.2.1	org.Hs.eg.db_3.10.0	latticeExtra_0.6-29
[184]	caTools_1.18.0	memoise_2.0.1	IRkernel_1.1
[187]	irlba_2.3.3	future.apply_1.6.0	ape_5.4

[]: