

scRNAseq_ETP_DN2

Ekaterina Takmakova

5 5 2021

Setup the Seurat Object

```
library(dplyr)
library(Seurat)
library(sctransform)
library(patchwork)
library(ggplot2)
library(xlsx)
library(openxlsx)

# Load the dataset
thymus.data <- Read10X(data.dir = "data/rep_1")

# Initialize the Seurat object with the raw (non-normalized data).
thymus <- CreateSeuratObject(counts = thymus.data, project = "ETP_DN3", min.cells = 3, min.features = 200)
thymus
```

```
## An object of class Seurat
## 14016 features across 5006 samples within 1 assay
## Active assay: RNA (14016 features, 0 variable features)
## 1 layer present: counts
```

QC and normalization of data

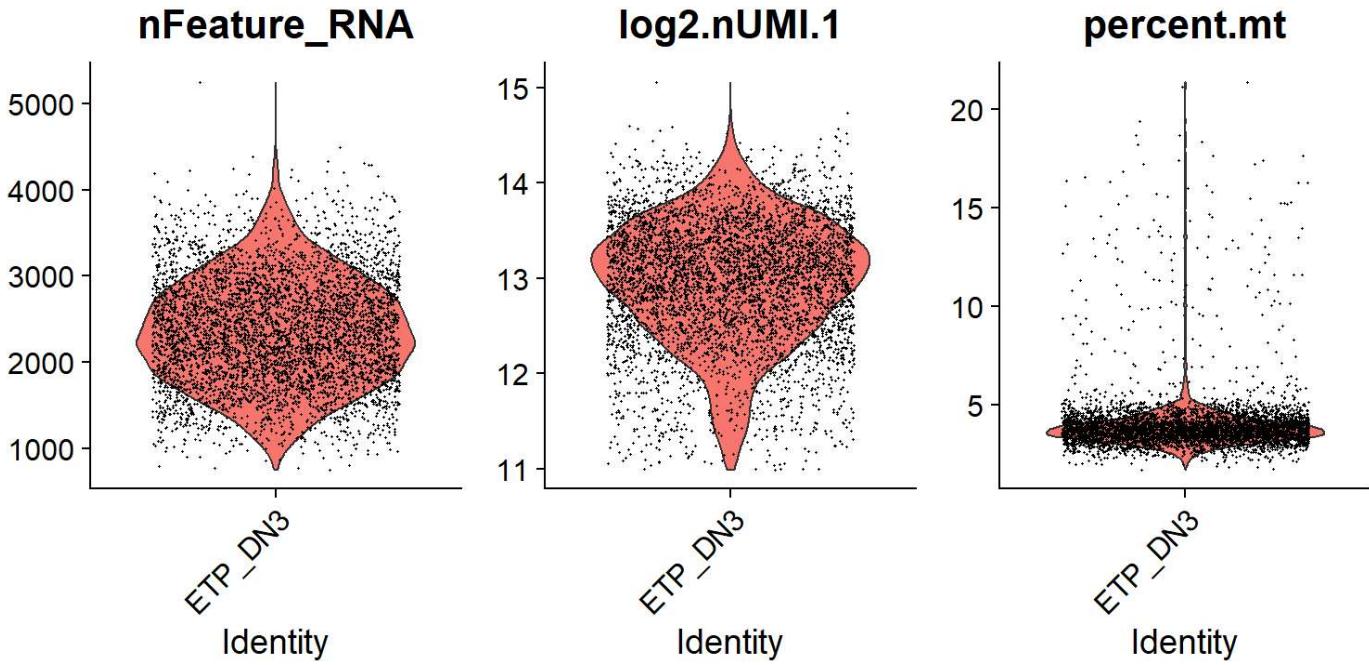
```
# Mitochondrial genes are marked as 'mt', 'mt-' for mouse

thymus[["percent.mt"]] <- PercentageFeatureSet(thymus, pattern = "^mt")

# Visualize QC metrics as a violin plot.
# nFeature_RNA is the number of genes detected in each cell
# nCount_RNA is the total number of molecules detected within a cell

thymus[["log2.nUMI.1"]] <- log2(thymus[["nCount_RNA"]]+1)

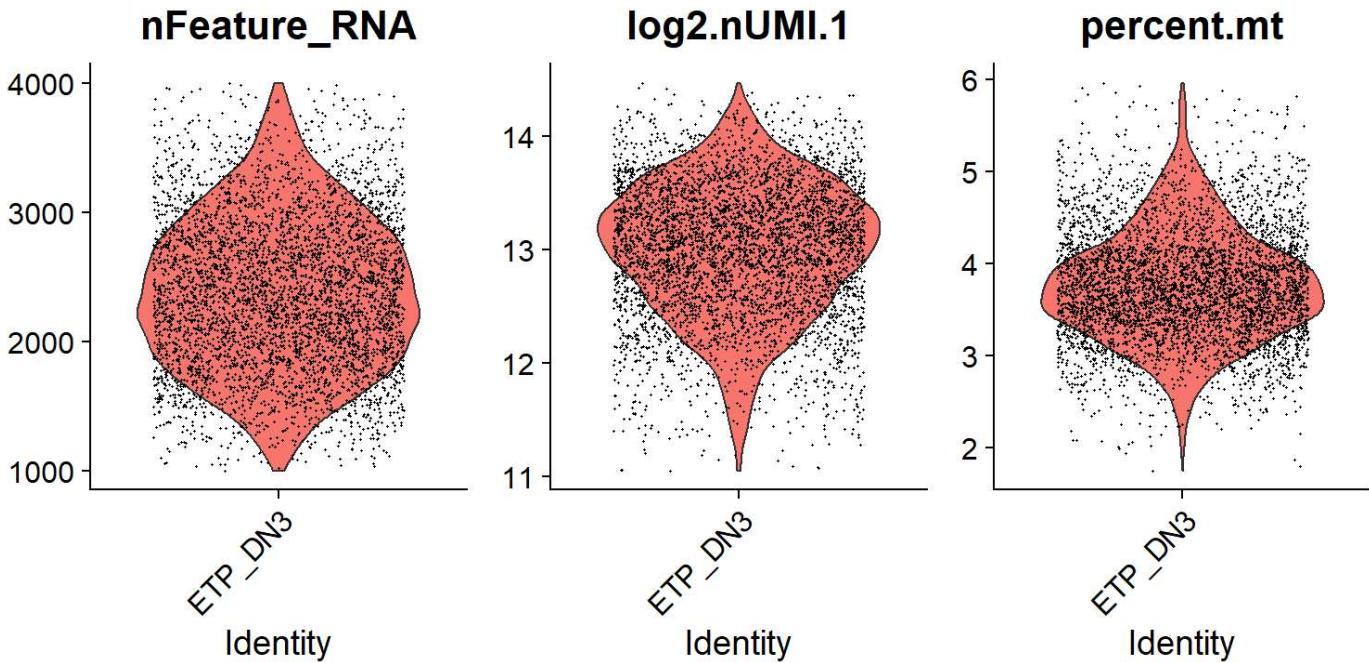
VlnPlot(thymus, features = c("nFeature_RNA", "log2.nUMI.1", "percent.mt"), ncol = 3)
```



Data selection

```
thymus <- subset(thymus, subset = nFeature_RNA > 1000 & nFeature_RNA < 4000 & nCount_RNA < 10000
00 & percent.mt < 6)

VlnPlot(thymus, features = c("nFeature_RNA", "log2.nUMI.1", "percent.mt"), ncol = 3)
```



Assign Cell-Cycle Scores

```

# A list of cell cycle markers from Tirosh et al 2015

m.s.genes <- read.table("data/s.genes.murine.txt")
m.s.genes <- as.character(m.s.genes)

m.g2m.genes <- read.table("data/g2m.genes.murine.txt")
m.g2m.genes <- as.character(m.g2m.genes)

# RNA assay - data normalization

thymus <- NormalizeData(thymus)

# Assign Cell-Cycle Scores

thymus <- CellCycleScoring(object = thymus, s.features = m.s.genes, g2m.features = m.g2m.genes)

```

Check localization of Cell cycle phases on UMAP

```

# RNA assay - data scaling

thymus <- FindVariableFeatures(thymus)
thymus <- ScaleData(thymus)

# Run PCA and UMAP

thymus <- RunPCA(thymus)
thymus <- RunUMAP(thymus, dims = 1:25)
thymus <- FindNeighbors(thymus, dims = 1:25)
thymus <- FindClusters(thymus, resolution = 0.9)

```

```

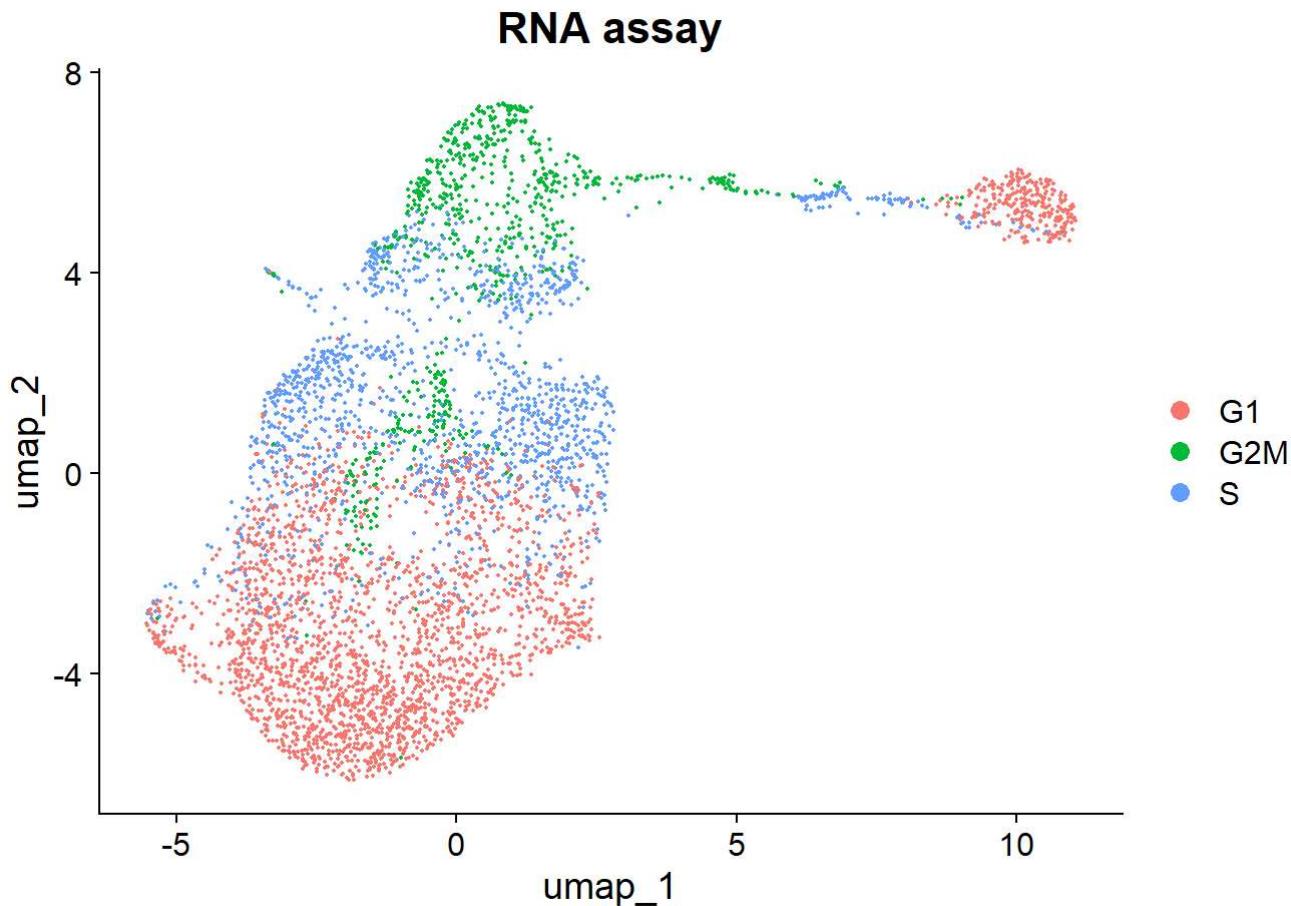
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 4786
## Number of edges: 181477
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7940
## Number of communities: 13
## Elapsed time: 0 seconds

```

```

DimPlot(thymus, reduction = 'umap', group.by = "Phase", label = FALSE, pt.size = 0.3, label.size = 5) +
  ggtitle("RNA assay")

```



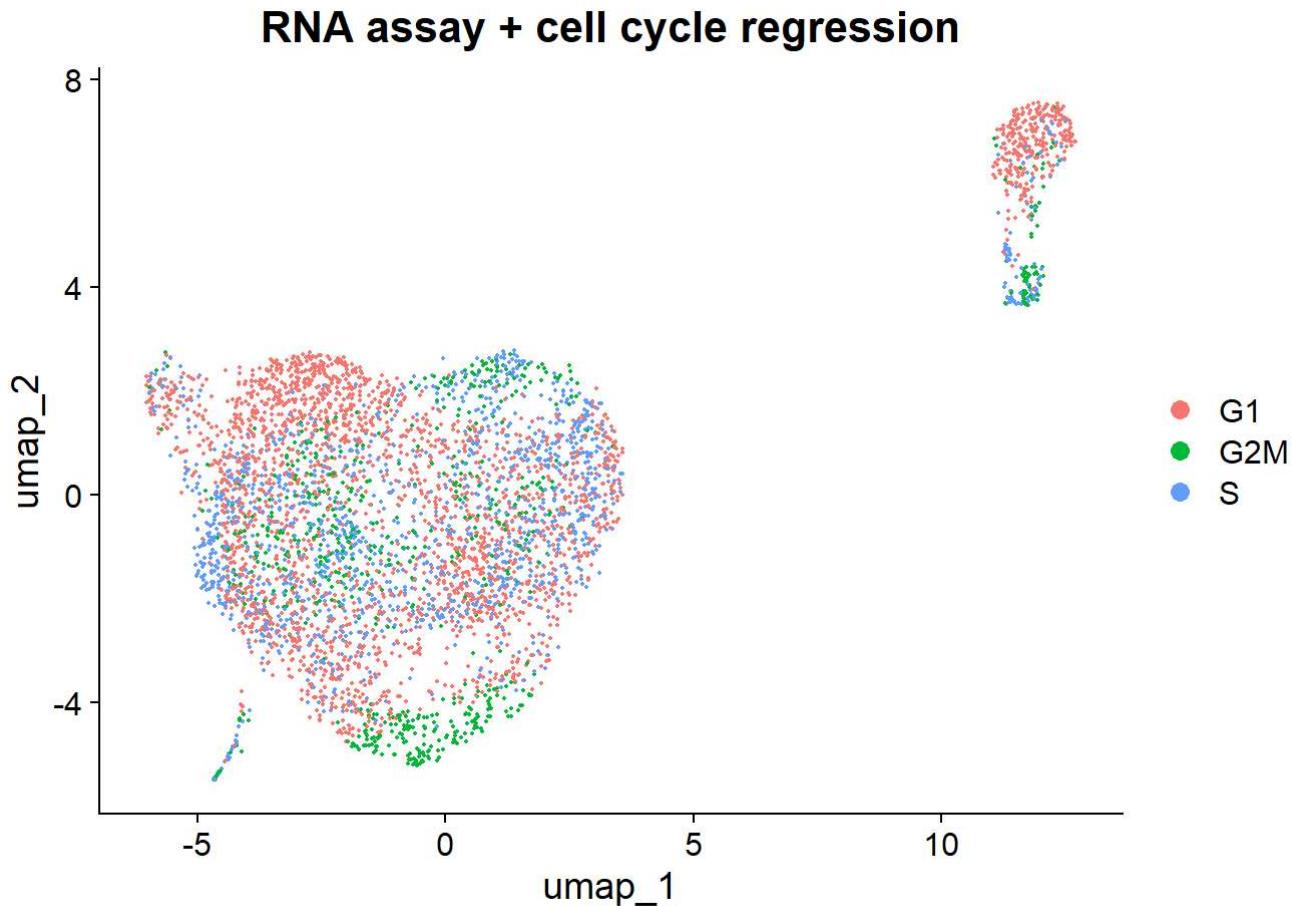
Regress out cell cycle scores during data scaling

```
thymus <- ScaleData(thymus, vars.to.regress = c("S.Score", "G2M.Score"))
```

```
thymus <- RunPCA(thymus)
thymus <- RunUMAP(thymus, dims = 1:25)
thymus <- FindNeighbors(thymus, dims = 1:25)
thymus <- FindClusters(thymus, resolution = 0.9)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 4786
## Number of edges: 176803
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7420
## Number of communities: 12
## Elapsed time: 0 seconds
```

```
DimPlot(thymus, reduction = 'umap', group.by = "Phase", label = FALSE, pt.size = 0.3, label.size = 5) +
  ggtitle("RNA assay + cell cycle regression")
```



Apply SCTransform normalization

```
# SCT assay - normalization and scaling of data + regressing out mitochondrial genes and cell cycle markers

thymus <- SCTransform(thymus, vars.to.regress = c("S.Score", "G2M.Score", "percent.mt"), return.only.var.genes = FALSE)

head(thymus[[1]])
```

```

##                                     orig.ident nCount_RNA nFeature_RNA percent.mt log2.nUMI.1
## AACCTGCAAGAGTCG-1      ETP_DN3     12125        2897   4.156701   13.56582
## AACCTGCAATAACGA-1      ETP_DN3     14239        3425   2.633612   13.79766
## AACCTGCAATAGCGG-1      ETP_DN3      7975        2349   4.426332   12.96145
## AACCTGCACTGAAGG-1      ETP_DN3     4538         1519   3.525782   12.14816
## AACCTGGTGGCAAAC-1      ETP_DN3     7118         2127   3.062658   12.79746
## AACCTGGTGGACCC-1      ETP_DN3    18758        3877   4.136902   14.19530
##                               S.Score   G2M.Score Phase RNA_snn_res.0.9
## AACCTGCAAGAGTCG-1  0.10151894  0.05677096   S       7
## AACCTGCAATAACGA-1 -0.06206695  0.31098596   G2M     4
## AACCTGCAATAGCGG-1  0.24829621 -0.10448688   S       2
## AACCTGCACTGAAGG-1 -0.23393964 -0.18205377   G1      0
## AACCTGGTGGCAAAC-1 -0.06937765 -0.21807493   G1      1
## AACCTGGTGGACCC-1 -0.22265380  0.43449641   G2M     2
##                               seurat_clusters nCount_SCT nFeature_SCT
## AACCTGCAAGAGTCG-1          7        9226        2871
## AACCTGCAATAACGA-1          4        9281        3294
## AACCTGCAATAGCGG-1          2        8268        2347
## AACCTGCACTGAAGG-1          0        7801        1558
## AACCTGGTGGCAAAC-1          1        8031        2122
## AACCTGGTGGACCC-1          2        9421        3189

```

Perform linear dimensional reduction (PCA)

```

thymus <- RunPCA(thymus, verbose = FALSE)

# Examine and visualize PCA results a few different ways
print(thymus[['pca']], dims = 1:10, nfeatures = 10)

```

```

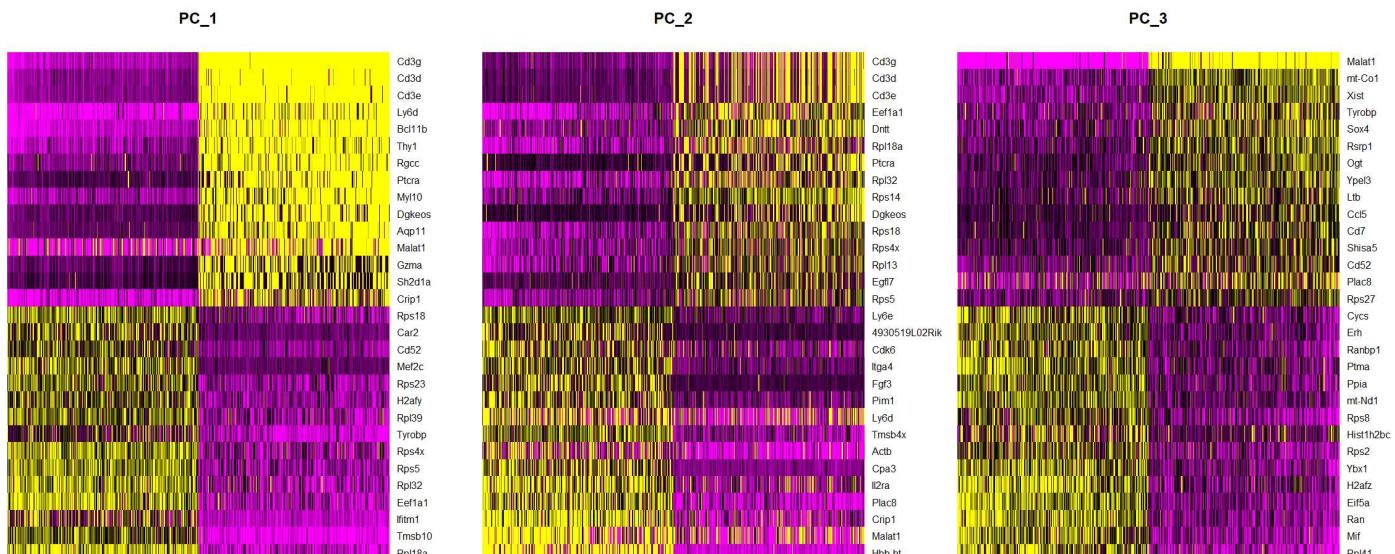
## PC_ 1
## Positive: Rpl18a, Tmsb10, Ifitm1, Eef1a1, Rpl32, Rps5, Rps4x, Tyrobp, Rpl39, H2afy
## Negative: Cd3g, Cd3d, Cd3e, Ly6d, Bcl11b, Thy1, Rgcc, Ptcra, Myl10, Dgkeos
## PC_ 2
## Positive: Hbb-bt, Malat1, Crip1, Plac8, Il2ra, Cpa3, Actb, Tmsb4x, Ly6d, Pim1
## Negative: Cd3g, Cd3d, Cd3e, Eef1a1, Dntt, Rpl18a, Ptcra, Rpl32, Rps14, Dgkeos
## PC_ 3
## Positive: Rpl41, Mif, Ran, Eif5a, H2afz, Ybx1, Rps2, Hist1h2bc, Rps8, mt-Nd1
## Negative: Malat1, mt-Co1, Xist, Tyrobp, Sox4, Rsrp1, Ogt, Ypel3, Ltb, Ccl5
## PC_ 4
## Positive: Plac8, Eef1a1, Ly6d, Rpl41, Hbb-bt, Lgals1, Crip1, Bcl11b, Rps18, Rpl32
## Negative: Malat1, Actb, Tubb5, Ncl, Cbx3, Set, Cd3d, Mif, Cd3g, Cd3e
## PC_ 5
## Positive: Malat1, Hbb-bt, Mif, Tyrobp, Tuba1b, Ncl, Hsp90aa1, Ifitm1, Ran, Eif5a
## Negative: Actb, Cbx3, Tubb5, Cdk6, Rpgrip1, Supt16, Slc39a1, Pim1, Notch2, Marf1
## PC_ 6
## Positive: Hbb-bt, Malat1, 2810417H13Rik, Hist1h2ap, Top2a, Tubb5, Rpl41, Rpl13, Rps19, Cd3d
## Negative: Isg15, Rtp4, Ifit1, Dntt, Lgals1, Plac8, Cenpa, Oasl2, Ifi27l2a, Hist1h2bc
## PC_ 7
## Positive: Hbb-bt, Cenpa, Ifitm1, Ccnb2, Cd3d, Cdc20, Ptcra, Ccnb1, Cdkn3, Knstrn
## Negative: 2810417H13Rik, Hist1h2ap, Top2a, Hist1h1b, Hist2h2aa1, Hist1h4i, Tk1, Hist1h2ae, Rrm2, H2afz
## PC_ 8
## Positive: Hbb-bt, Isg15, Rtp4, Ifit1, 2810417H13Rik, Oasl2, Rsad2, Usp18, Irf7, Gm4955
## Negative: Malat1, Crip1, Ube2c, Ifitm1, Il2ra, Mta3, Cpa3, Cenpa, Bcl2, Bcl11b
## PC_ 9
## Positive: Isg15, Ifit1, Rtp4, Malat1, Usp18, Oasl2, Gm4955, Rsad2, Crip1, Irf7
## Negative: Hbb-bt, Lgals1, Mpo, Hist1h2bc, 2810417H13Rik, Ccl9, Prtn3, Car2, Ctsg, Hmgb2
## PC_ 10
## Positive: Hbb-bt, Crip1, Dntt, Ly6d, Egfl7, Tpm4, Thy1, Gzma, Gm43643, Cd34
## Negative: Prtn3, Ctsg, Mpo, Ccl9, Ifitm1, Ms4a3, Elane, F630028010Rik, Ptcra, Cebpa

```

```

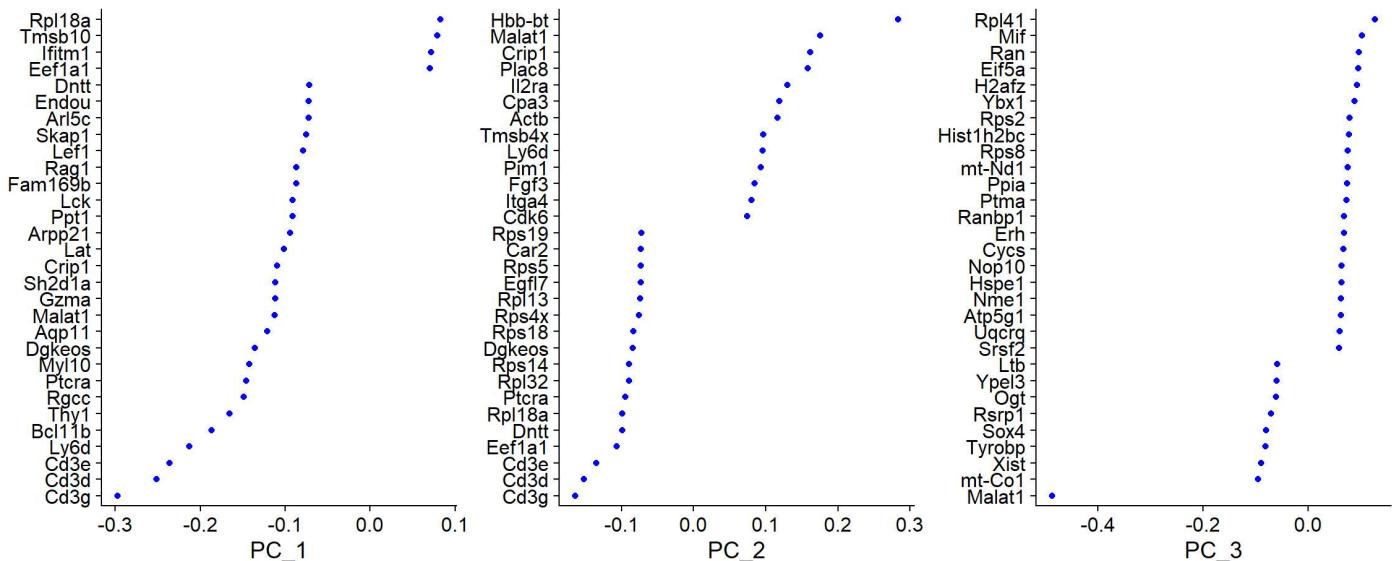
DimHeatmap(thymus, dims = 1:3, cells = 500, balanced = TRUE) + theme(text = element_text(size = 90))

```

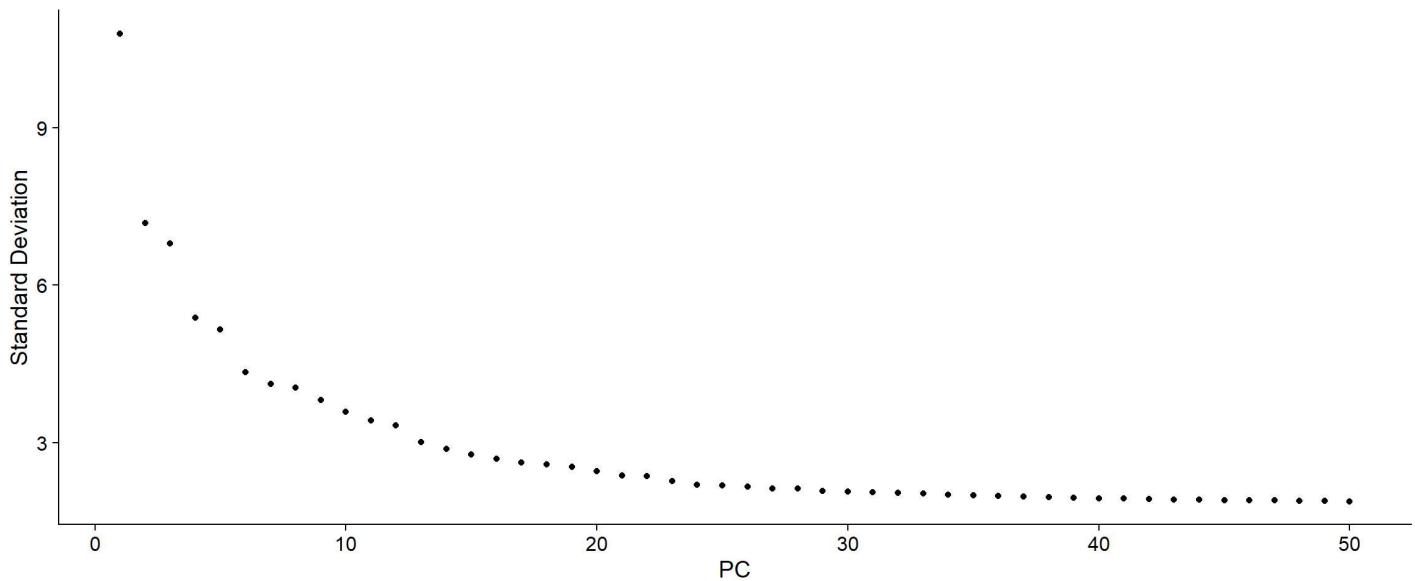


```
## NULL
```

```
VizDimLoadings(thymus, dims = 1:3, reduction = "pca", ncol = 3)
```



```
# Determine the 'dimensionality' of the dataset  
ElbowPlot(thymus, ndims = 50)
```



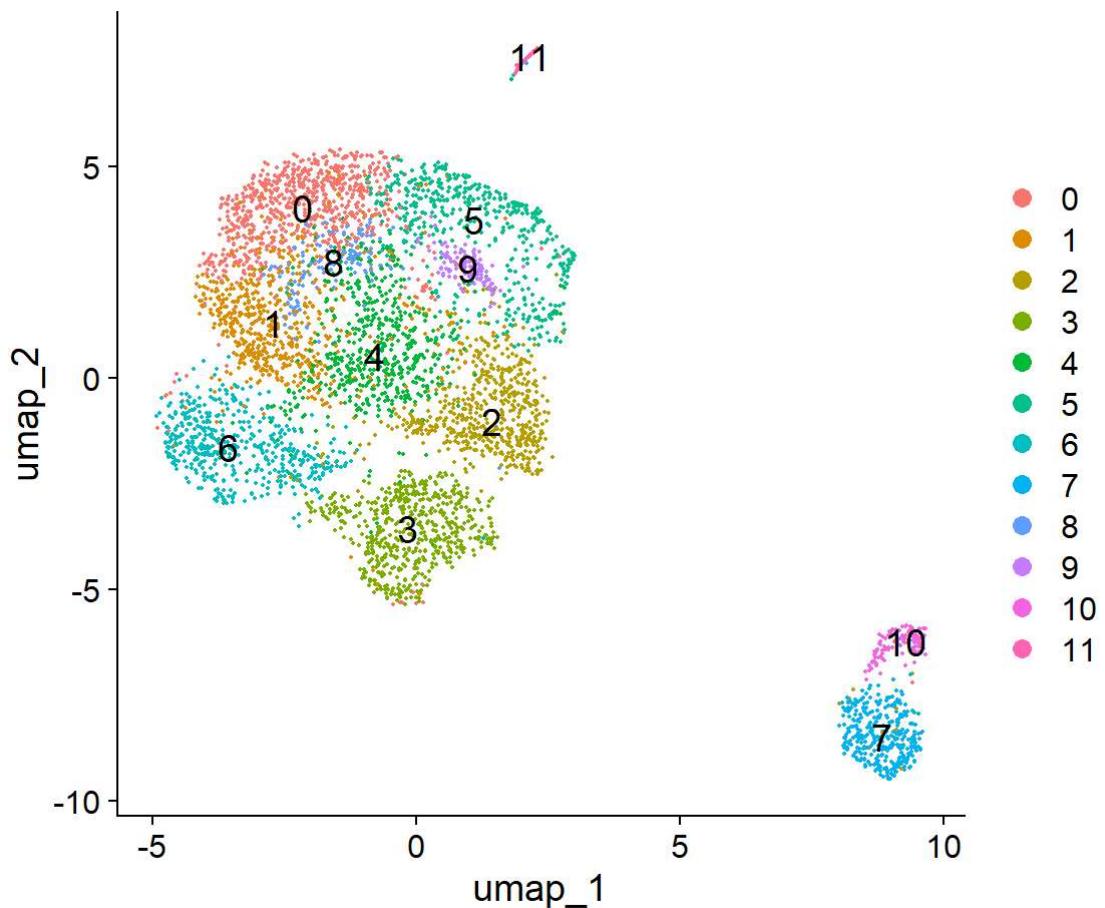
Cluster the cells, run non-linear dimensional reduction (UMAP)

```
DefaultAssay(thymus) <- "SCT"

thymus <- RunUMAP(thymus, dims = 1:30)
thymus <- FindNeighbors(thymus, dims = 1:30)
thymus <- FindClusters(thymus, resolution = 0.9)
```

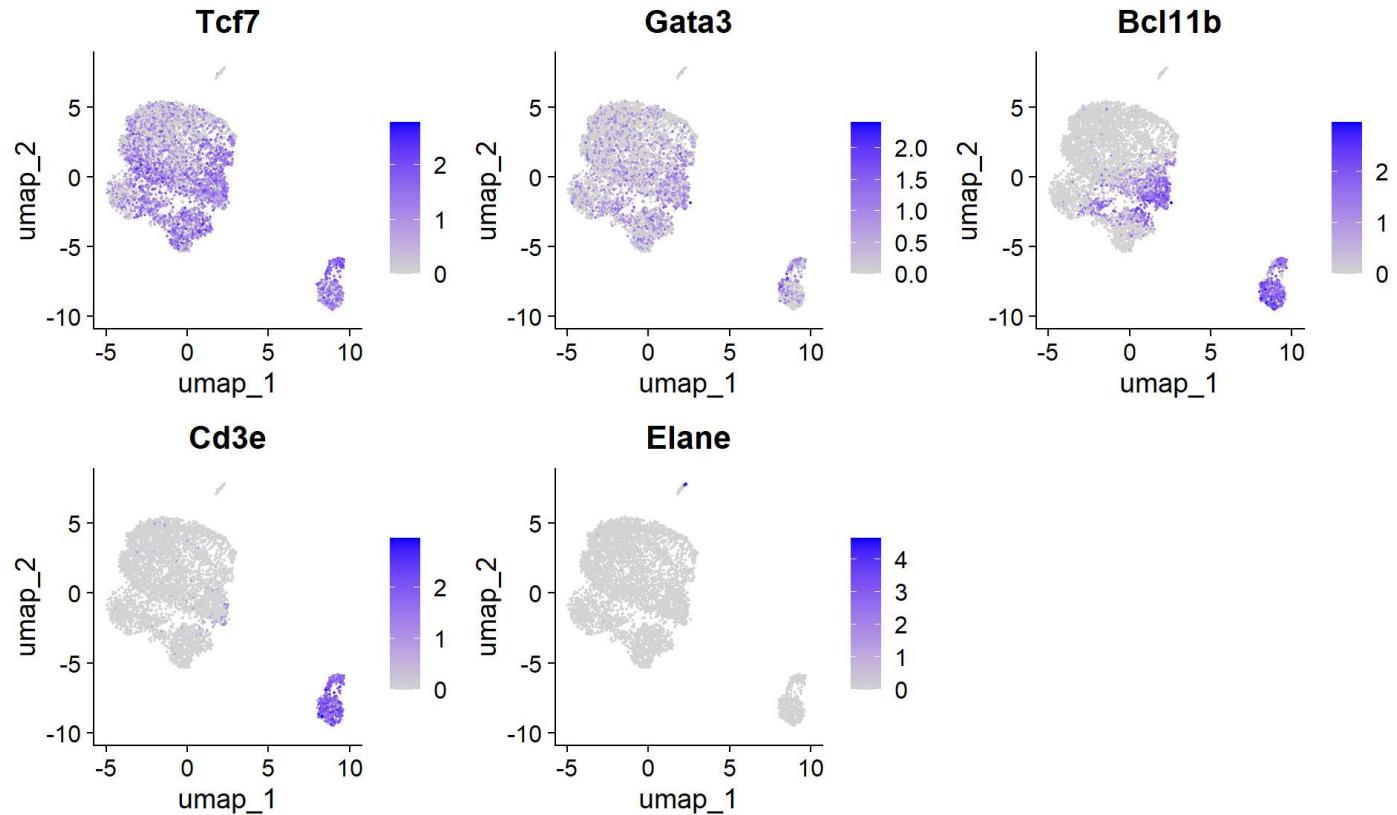
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 4786
## Number of edges: 183866
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7750
## Number of communities: 12
## Elapsed time: 0 seconds
```

```
DimPlot(thymus, reduction = 'umap', label = TRUE, pt.size = 0.3, label.size = 5)
```



Check the origin of the cells using typical genes expressed in thymocytes

```
FeaturePlot(thymus, features = c("Tcf7", "Gata3", "Bcl11b", "Cd3e", "Elane"), ncol = 3)
```



```
# Small group of cells do not express Tcf7, Gata3, Bcl11b, CD3, but do express Elane
```

The ELANE gene provides instructions for making a protein called neutrophil elastase. Cells expressing Elane are from myeloid fraction. Subset cells with absence of Elane expression.

```
thymus_noGrP <- subset(thymus, Elane == 0)  
thymus_noGrP
```

```
## An object of class Seurat  
## 27383 features across 4753 samples within 2 assays  
## Active assay: SCT (13367 features, 3000 variable features)  
## 3 layers present: counts, data, scale.data  
## 1 other assay present: RNA  
## 2 dimensional reductions calculated: pca, umap
```

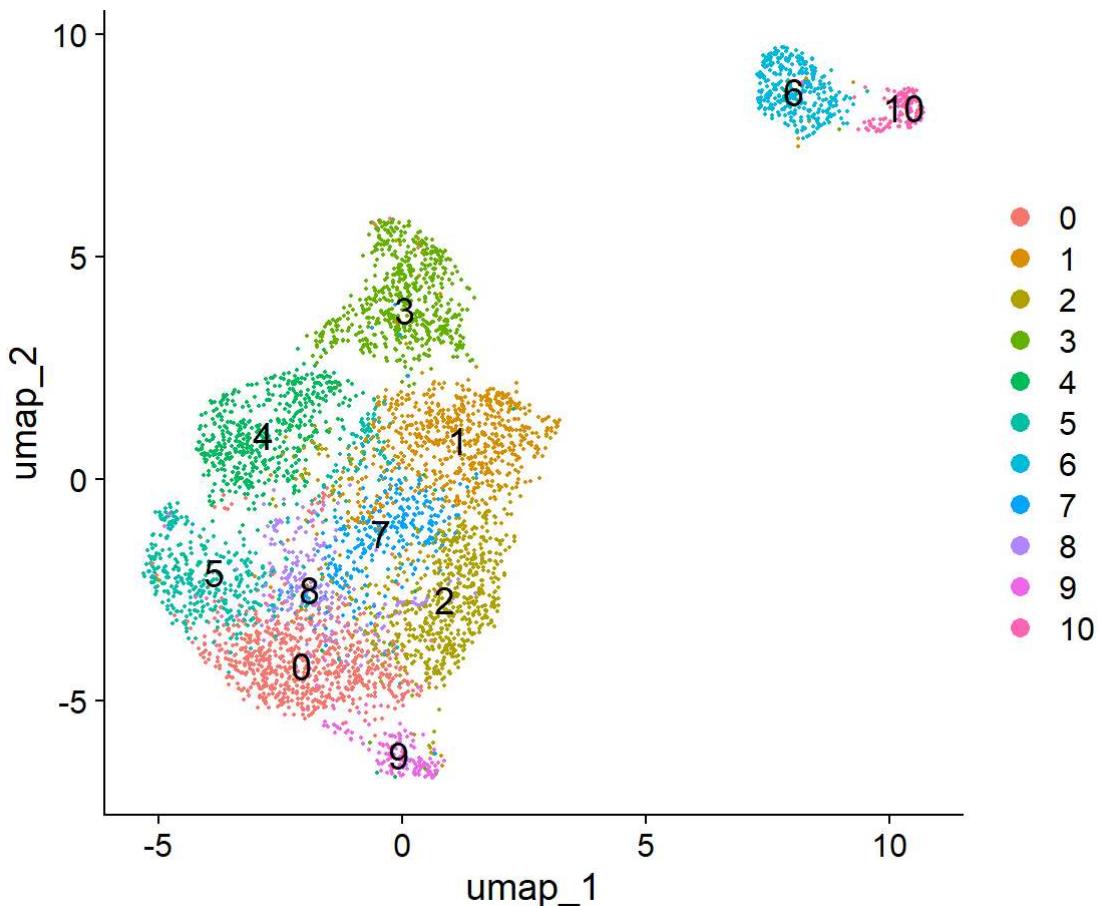
Repeat UMAP and clustering on pure thimic population

```
#Number of dimensions were chosen to achieve good separation between ETP (Flt3+) cells and other  
s
```

```
thymus_noGrP <- RunUMAP(thymus_noGrP, dims = 1:24)  
thymus_noGrP <- FindNeighbors(thymus_noGrP, dims = 1:12)  
thymus_noGrP <- FindClusters(thymus_noGrP, resolution = 0.9)
```

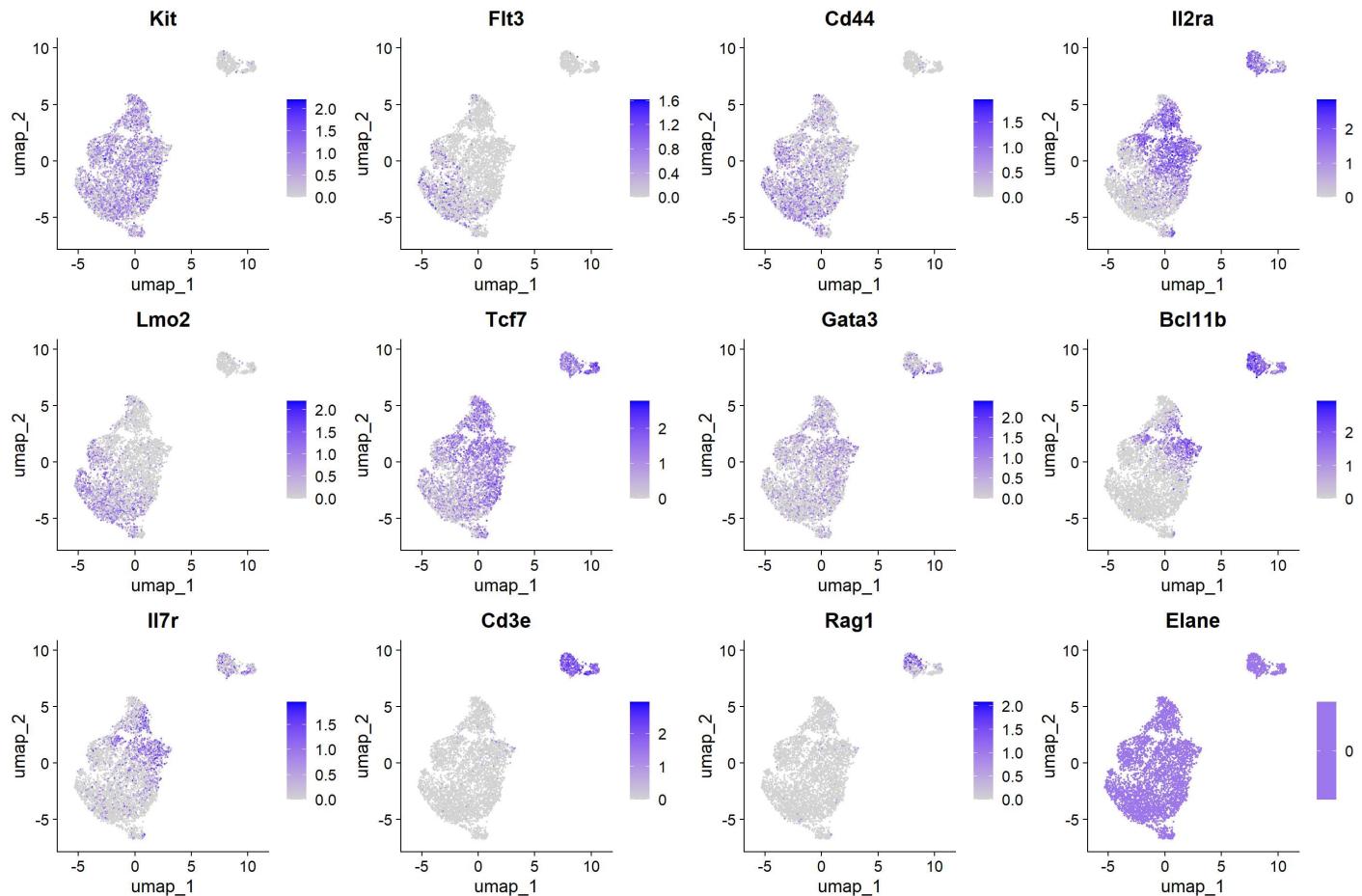
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck  
##  
## Number of nodes: 4753  
## Number of edges: 159786  
##  
## Running Louvain algorithm...  
## Maximum modularity in 10 random starts: 0.7857  
## Number of communities: 11  
## Elapsed time: 0 seconds
```

```
DimPlot(thymus_noGrP, reduction = 'umap', label = TRUE, pt.size = 0.3, label.size = 5)
```

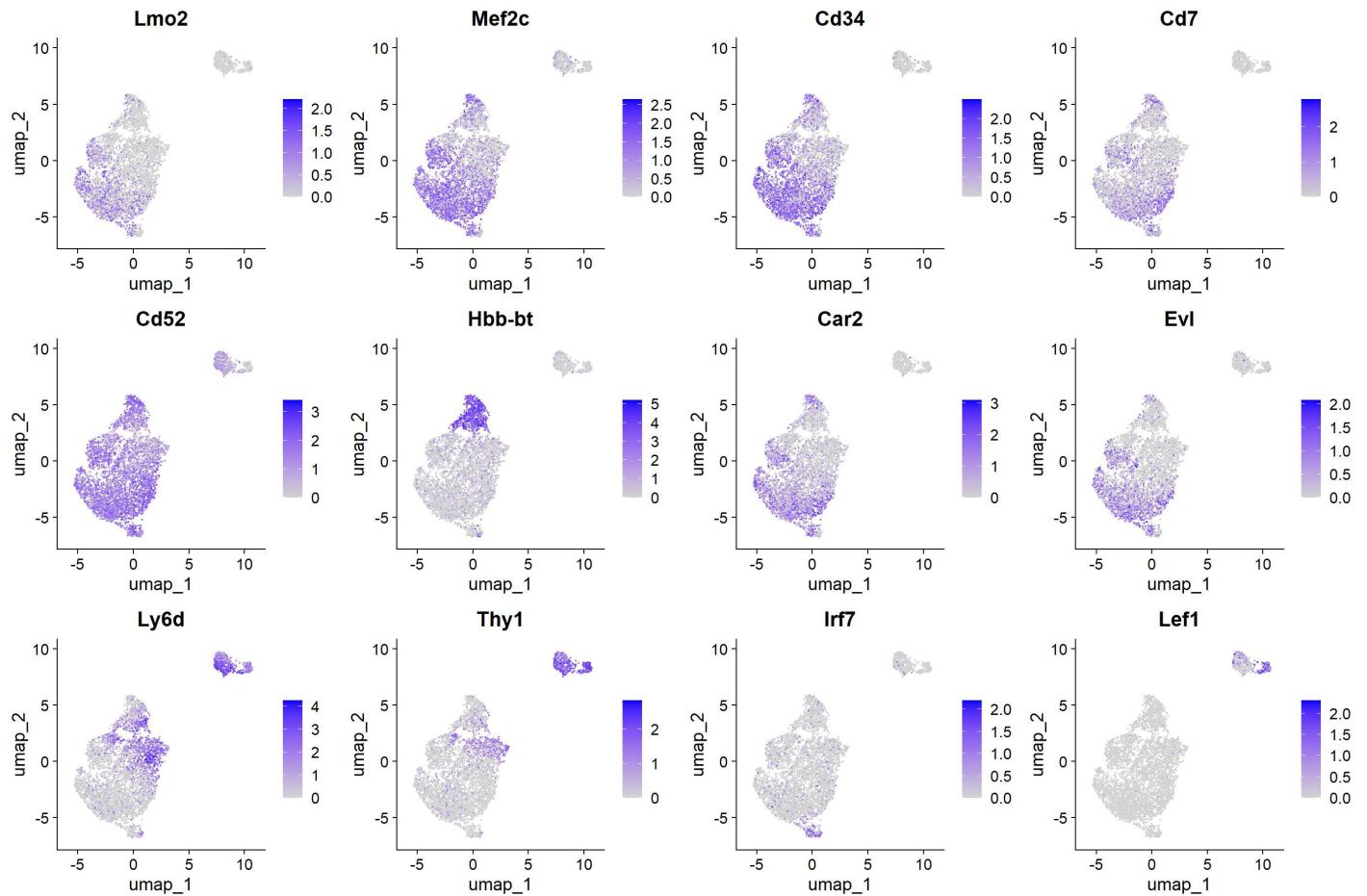


Visualize specific markers to identify clusters as cell populations

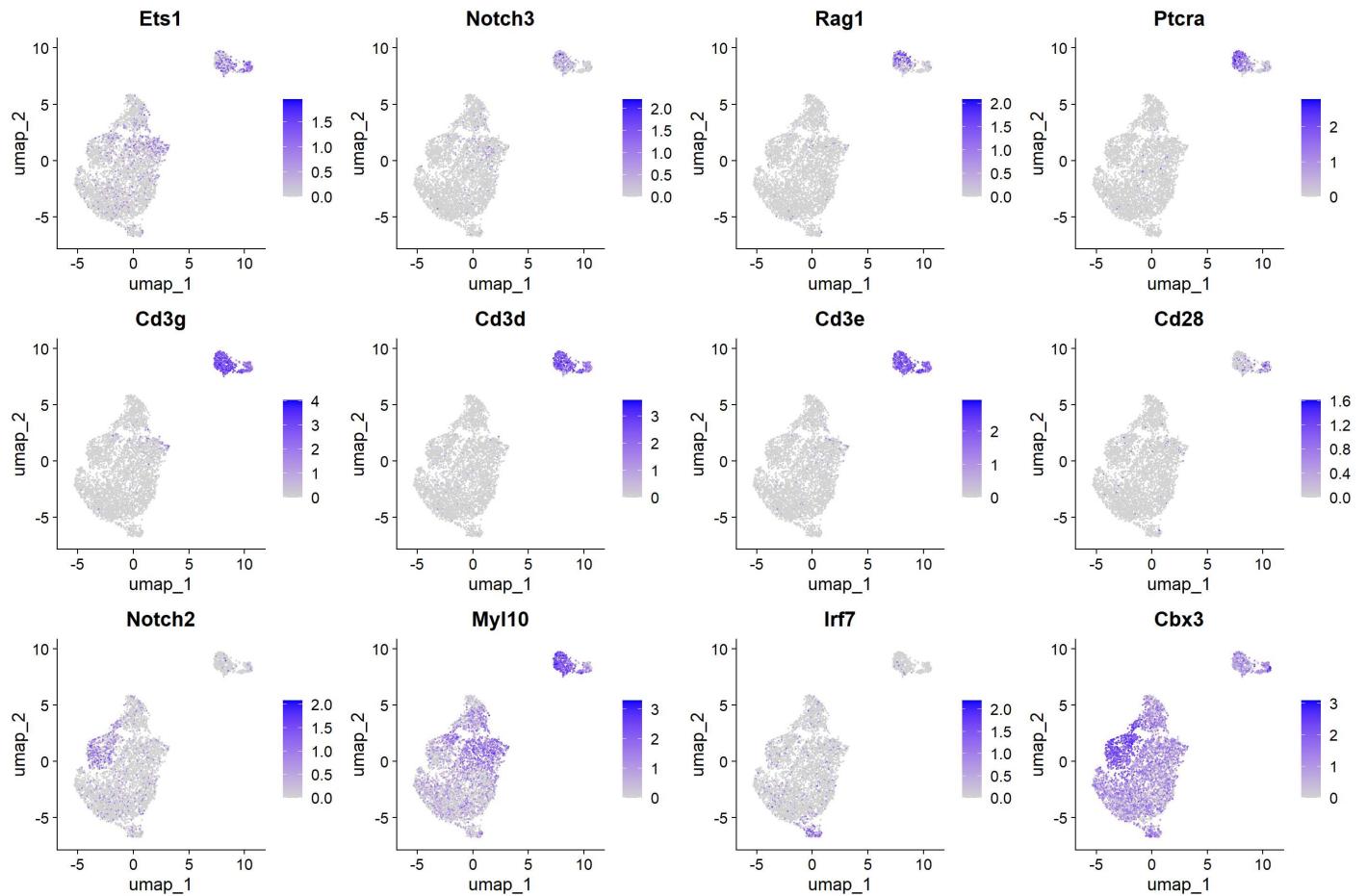
```
FeaturePlot(thymus_noGrP, features = c("Kit", "Flt3", "Cd44", "Il2ra", "Lmo2", "Tcf7", "Gata3",  
"Bcl11b", "Il7r", "Cd3e", "Rag1", "Elane"), ncol = 4)
```



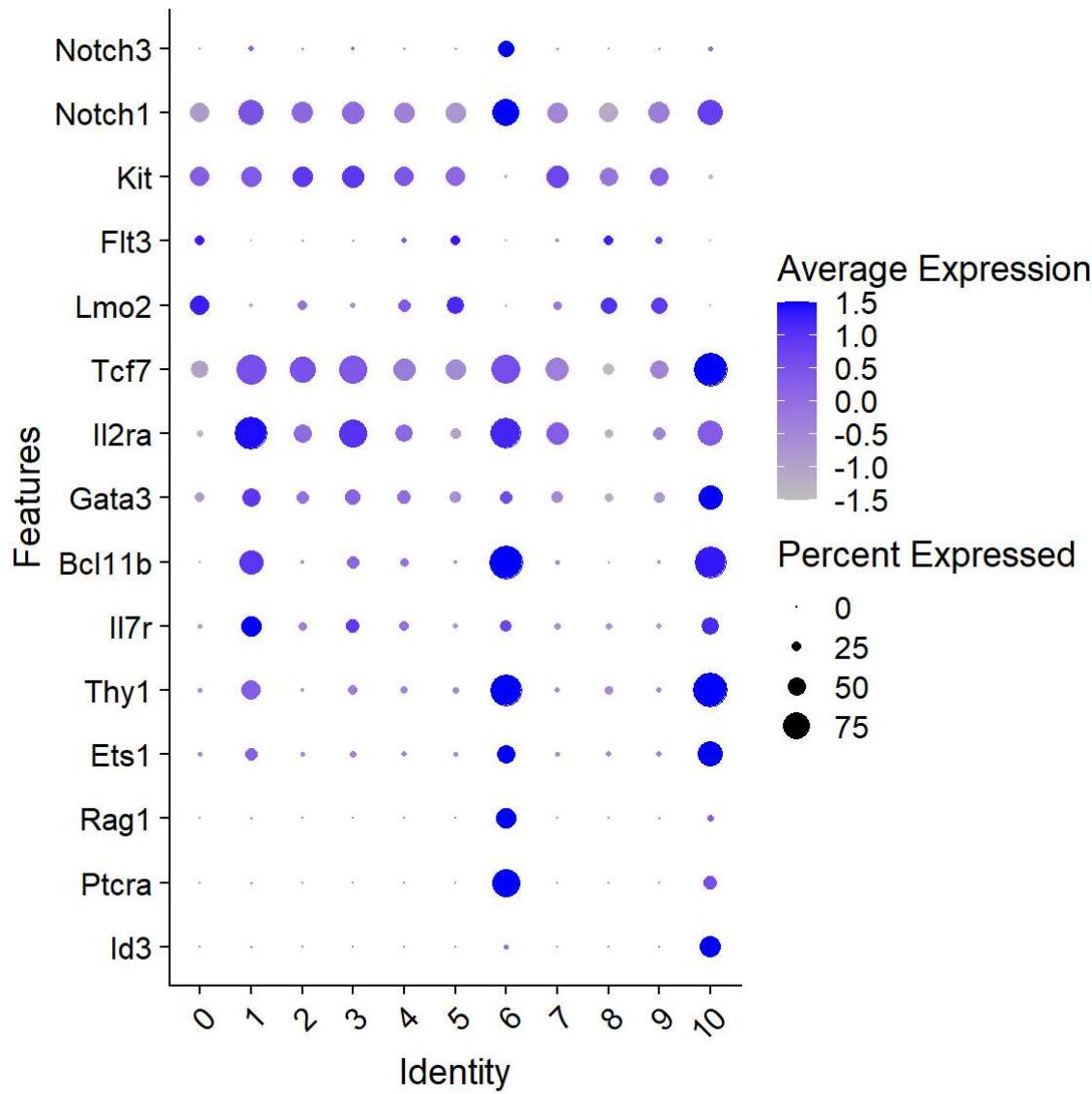
```
FeaturePlot(thymus_noGrP, features = c("Lmo2", "Mef2c", "Cd34", "Cd7", "Cd52", "Hbb-bt", "Car2",  
"Evl", "Ly6d", "Thy1", "Irf7", "Lef1"), ncol = 4)
```



```
FeaturePlot(thymus_noGrP, features = c("Ets1", "Notch3", "Rag1", "Ptcr", "lef1", "Cd3g", "Cd3d", "Cd3e", "Cd28", "Notch2", "Myl10", "Irf7", "Cbx3"), ncol = 4)
```



```
DotPlot(thymus_noGrP, features = c("Id3", "Ptcra", "Rag1", "Ets1", "Thy1", "Il17r", "Bcl11b", "Gata3", "Il2ra", "Tcf7", "Lmo2", "Flt3", "Kit", "Notch1", "Notch3"), cols = c("grey", "blue"), co  
l.min = -1.5, col.max = 1.5) + RotatedAxis() + coord_flip()
```



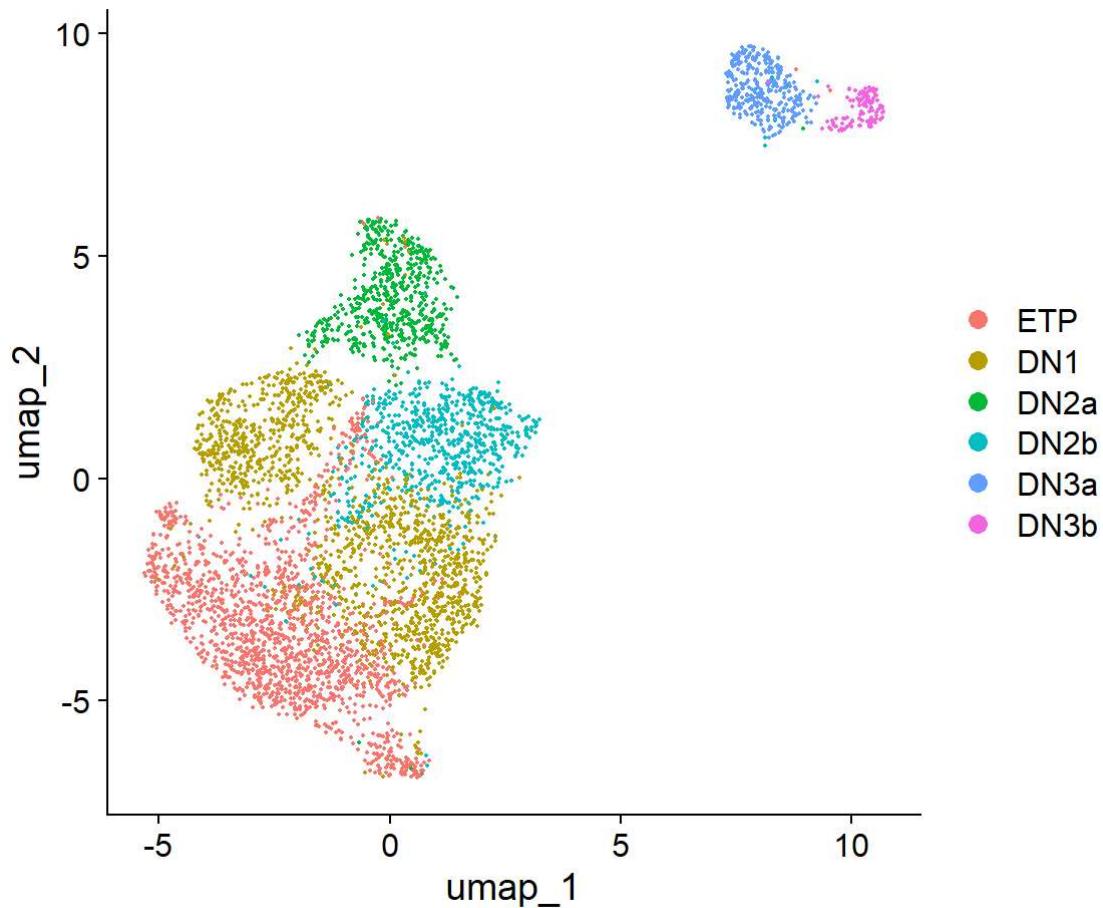
Cluster annotation

```
# ETP cluster - Flt3+ Tcf7_Low
# DN1 cluster - Bcl11b- Lmo2_Low Tcf7+
# DN2a cluster - Bcl11b_Low Lmo2- IL7r_Low
# DN2b cluster - Bcl11b+ IL7r+
# DN3a cluster - Rag1+ Notch3+ Kit-
# DN3b cluster - Id3+ Kit-

thymus_noGrP <- RenameIdents(thymus_noGrP, "0" = "ETP", "1" = "DN2b", "2" = "DN1", "3" = "DN2a",
"4" = "DN1", "5" = "ETP", "6" = "DN3a", "7" = "DN1", "8" = "ETP", "9" = "ETP", "10" = "DN3b")

thymus_noGrP@active.ident <- factor(x = thymus_noGrP@active.ident, levels = c("ETP", "DN1", "DN2a",
"DN2b", "DN3a", "DN3b"))

DimPlot(thymus_noGrP, reduction = 'umap', label = FALSE, pt.size = 0.3, label.size = 5)
```

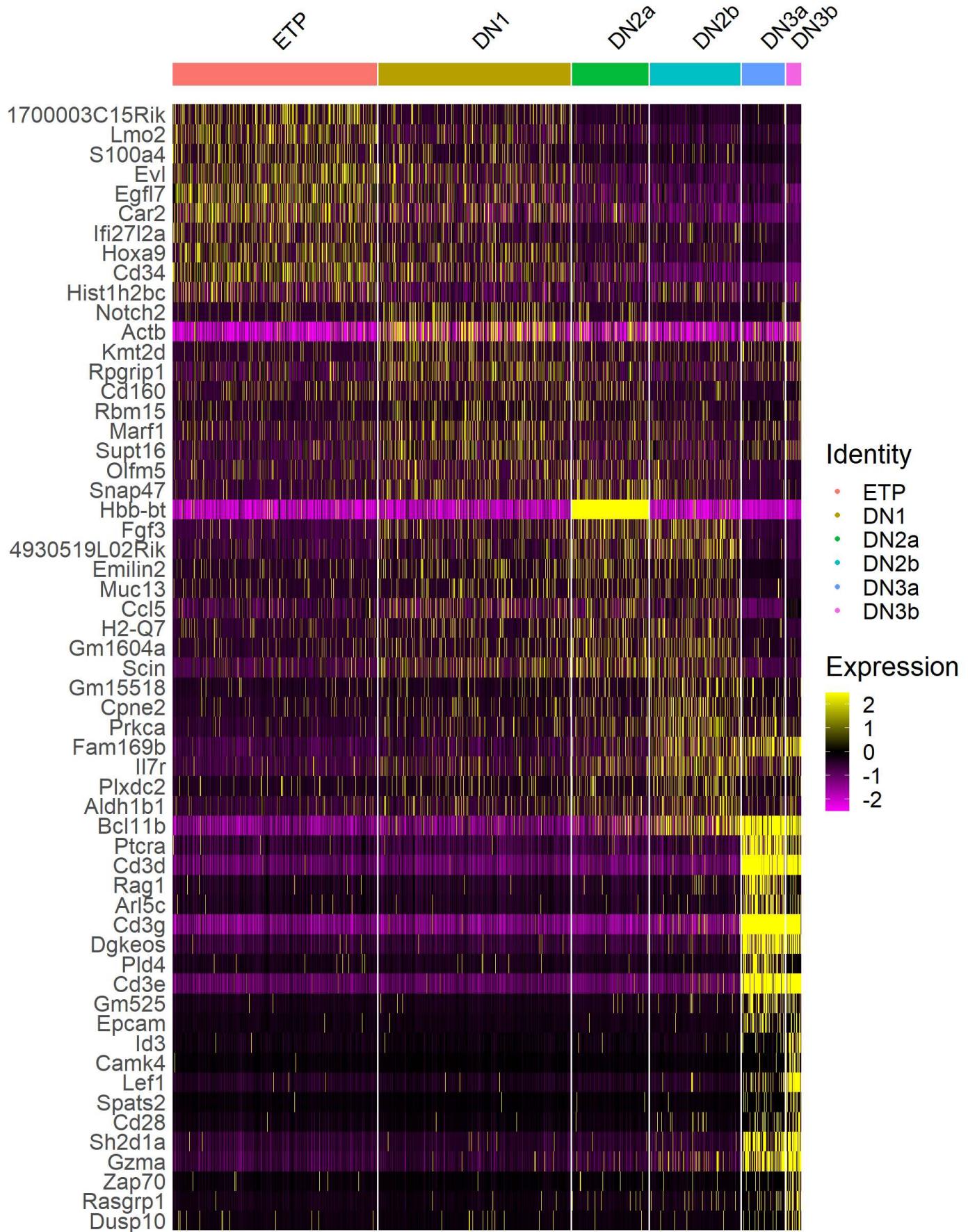


Finding differentially expressed features (cluster biomarkers)

```
thymus_noGrP.markers <- FindAllMarkers(thymus_noGrP, only.pos = TRUE, min.pct = 0.25, logfc.thre
shold = 0.25)
thymus_noGrP.markers %>% group_by(cluster) %>% slice_max(n = 10, order_by = avg_log2FC) -> top10
top10
```

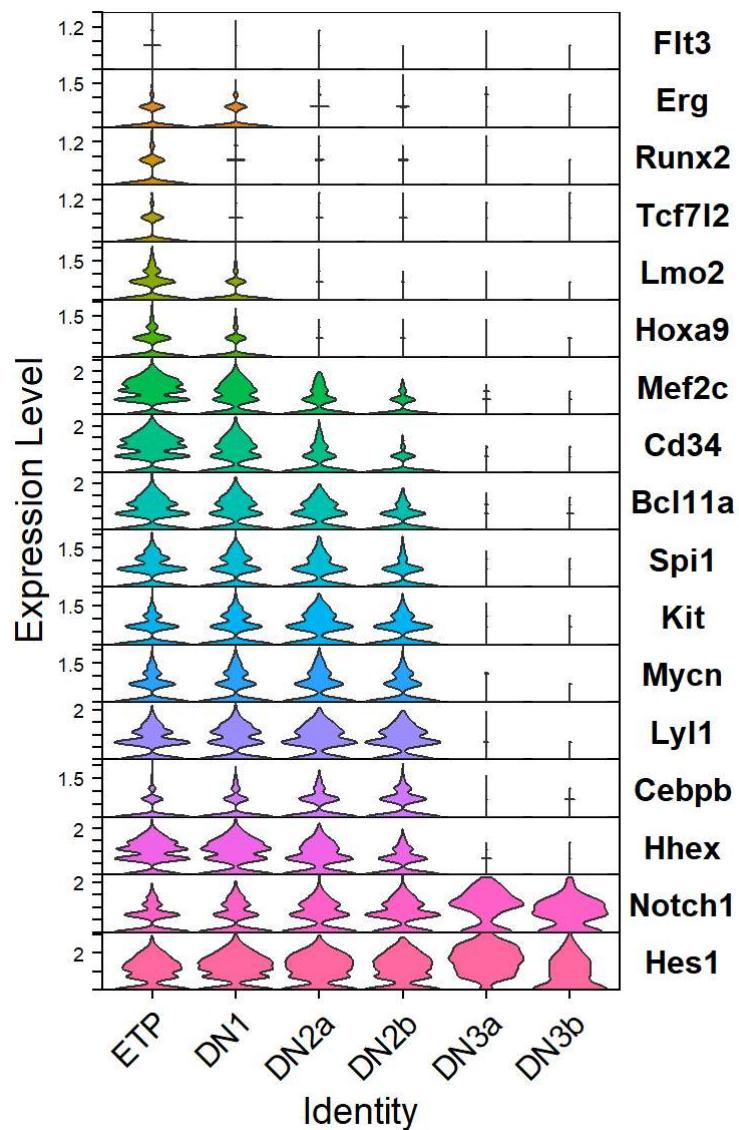
```
## # A tibble: 60 × 7
## # Groups:   cluster [6]
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##       <dbl>     <dbl> <dbl>    <dbl>    <dbl> <fct>  <chr>
## 1 1.35e-100    2.06  0.307  0.072  1.80e- 96 ETP    1700003C15Rik
## 2 2.18e-138    1.85  0.492  0.156  2.91e-134 ETP    Lmo2
## 3 7.48e- 67    1.54  0.294  0.098  1.00e- 62 ETP    S100a4
## 4 7.66e-128    1.49  0.546  0.2    1.02e-123 ETP    Evl
## 5 3.38e-157    1.47  0.743  0.377  4.52e-153 ETP    Egfl7
## 6 4.75e-140    1.33  0.713  0.337  6.35e-136 ETP    Car2
## 7 1.82e- 36    1.30  0.297  0.147  2.43e- 32 ETP    Ifi27l2a
## 8 2.39e- 65    1.28  0.416  0.188  3.19e- 61 ETP    Hoxa9
## 9 1.95e-164    1.24  0.832  0.456  2.61e-160 ETP    Cd34
## 10 1.63e- 44   1.23  0.408  0.225  2.17e- 40 ETP    Hist1h2bc
## # i 50 more rows
```

```
DoHeatmap(thymus_noGrP, features = top10$gene) + theme(text = element_text(size = 20))
```

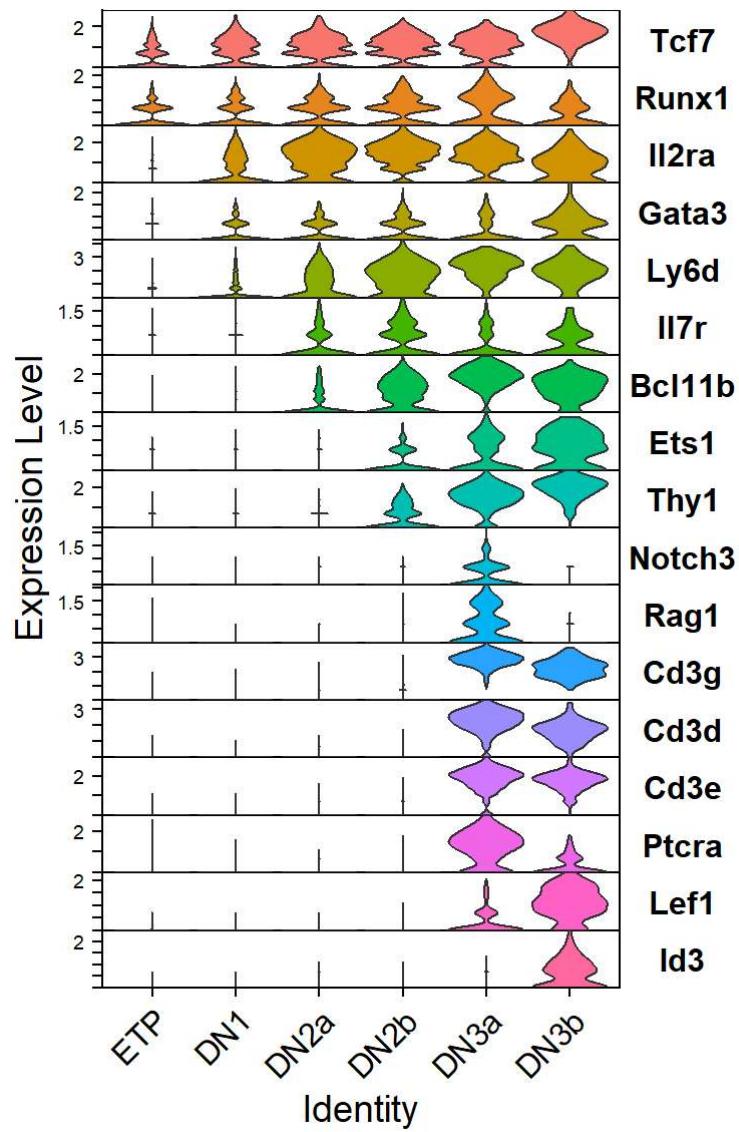


Visualize specific markers

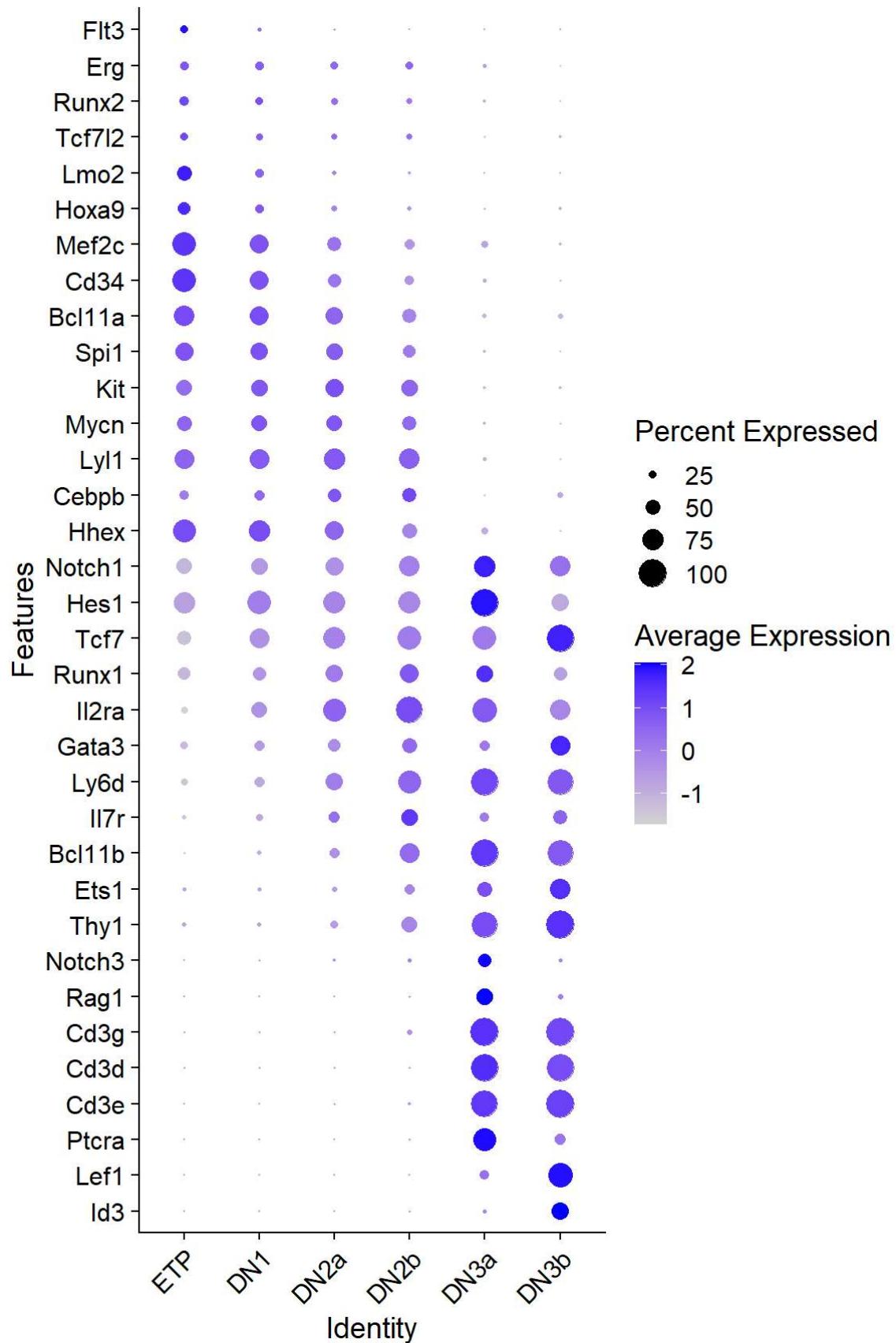
```
VlnPlot(thymus_noGrP, features = c("Flt3", "Erg", "Runx2", "Tcf7l2", "Lmo2", "Hoxa9", "Mef2c",  
"Cd34", "Bcl11a", "Spi1", "Kit", "Mycn", "Lyl1", "Cebpb", "Hhex", "Notch1", "Hes1"), stack = TRUE,  
flip = TRUE) + NoLegend()
```



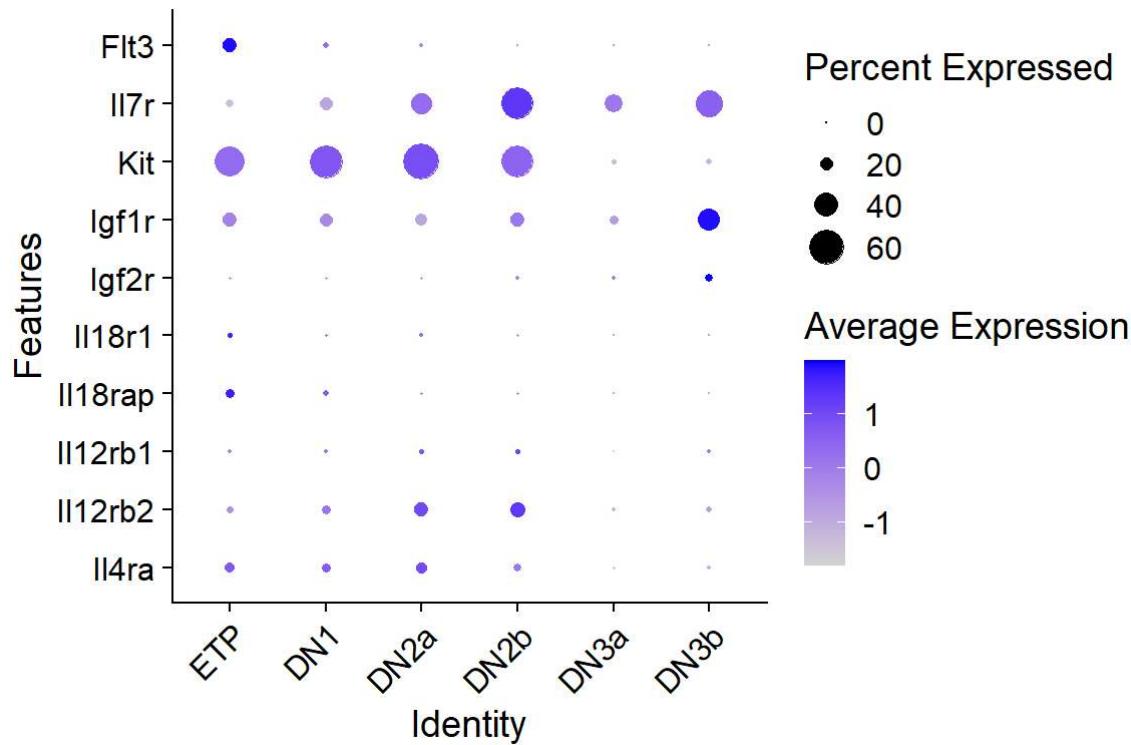
```
VlnPlot(thymus_noGrP, features = c("Tcf7", "Runx1", "Il2ra", "Gata3", "Ly6d", "Il7r", "Bcl11b",  
"Ets1", "Thy1", "Notch3", "Rag1", "Cd3g", "Cd3d", "Cd3e", "Ptcr", "Lef1", "Id3"), stack = TRUE,  
flip = TRUE) + NoLegend()
```



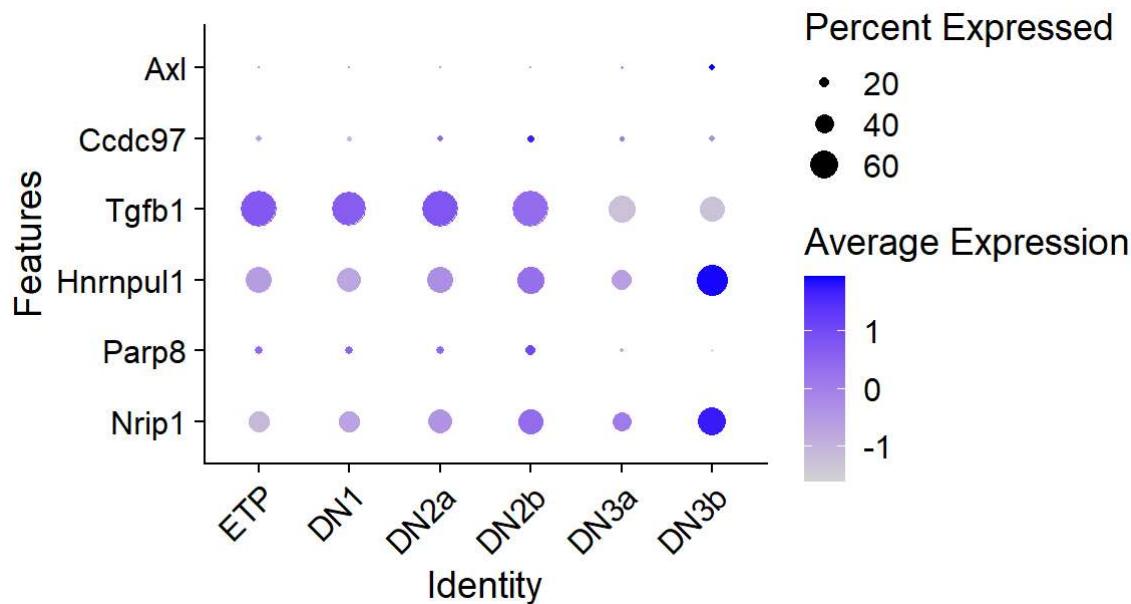
```
DotPlot(thymus_noGrP, features = c("Id3", "Lef1", "Ptcra", "Cd3e", "Cd3d", "Cd3g", "Rag1", "Notch3", "Thy1", "Ets1", "Bcl11b", "Il7r", "Ly6d", "Gata3", "Il2ra", "Runx1", "Tcf7", "Hes1", "Notch1", "Hhex", "Cebpb", "Lyl1", "Mycn", "Kit", "Spi1", "Bcl11a", "Cd34", "Mef2c", "Hoxa9", "Lmo2", "Tcf7l2", "Runx2", "Erg", "Flt3")) + RotatedAxis() + coord_flip()
```



```
DotPlot(thymus_noGrP, features = c("Il4ra", "Il12rb2", "Il12rb1", "Il18rap", "Il18r1", "Igf2r", "Igf1r", "Kit", "Il7r", "Flt3")) + RotatedAxis() + coord_flip()
```



```
DotPlot(thymus_noGrP, features = c("Nrip1", "Parp8", "Hnrnpull1", "Tgfb1", "Ccdc97", "Axl")) + Rota
tedAxis() + coord_flip()
```



```
AverageExpression(thymus_noGrP, assays = "SCT", slot = "data", features = c("Il7r", "Nrip1", "Pa
rp8", "Hnrnpull1", "Tgfb1", "Ccdc97", "Axl"))
```

```
## $SCT
## 7 x 6 sparse Matrix of class "dgCMatrix"
##           ETP      DN1      DN2a      DN2b      DN3a      DN3b
## I17r    0.14002558 0.27074830 0.61578045 0.99421965 0.54128440 0.69230769
## Nrip1   0.60102302 0.65850340 0.70154374 0.82080925 0.77675841 1.05128205
## Parp8   0.17838875 0.18027211 0.17667238 0.22109827 0.08562691 0.03418803
## Hnrnpull1 0.83056266 0.80884354 0.85763293 0.92485549 0.82568807 1.11111111
## Tgfb1   1.73785166 1.68707483 1.75814751 1.56502890 0.93272171 0.92307692
## Ccdc97  0.10485934 0.09659864 0.12178388 0.14161850 0.11620795 0.11111111
## Axl     0.02429668 0.02244898 0.01543739 0.01300578 0.04281346 0.10256410
```

Session Info

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22621)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: Europe/Kiev
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets  methods   base
##
## other attached packages:
## [1] openxlsx_4.2.5.2    xlsx_0.6.5        ggplot2_3.4.4      patchwork_1.1.3
## [5] sctransform_0.4.1   Seurat_5.0.1       SeuratObject_5.0.1 sp_2.1-2
## [9] dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] RColorBrewer_1.1-3          rstudioapi_0.15.0
##  [3] jsonlite_1.8.8              magrittr_2.0.3
##  [5] spatstat.utils_3.0-4        farver_2.1.1
##  [7] rmarkdown_2.25               zlibbioc_1.48.0
##  [9] vctrs_0.6.5                ROCR_1.0-11
## [11] DelayedMatrixStats_1.24.0   spatstat.explore_3.2-5
## [13] RCurl_1.98-1.13            S4Arrays_1.2.0
## [15] htmltools_0.5.7             SparseArray_1.2.2
## [17] sass_0.4.7                 parallelly_1.36.0
## [19] KernSmooth_2.23-22         bslib_0.6.1
## [21] htmlwidgets_1.6.3           ica_1.0-3
## [23] plyr_1.8.9                 plotly_4.10.3
## [25] zoo_1.8-12                 cachem_1.0.8
## [27] igraph_1.5.1               mime_0.12
## [29] lifecycle_1.0.4             pkgconfig_2.0.3
## [31] Matrix_1.6-4               R6_2.5.1
## [33] fastmap_1.1.1              GenomeInfoDbData_1.2.11
## [35] MatrixGenerics_1.14.0       fitdistrplus_1.1-11
## [37] future_1.33.0              shiny_1.8.0
## [39] digest_0.6.33              colorspace_2.1-0
## [41] S4Vectors_0.40.2            tensor_1.5
## [43] RSpectra_0.16-1             irlba_2.3.5.1
## [45] GenomicRanges_1.54.1        labeling_0.4.3
## [47] progressr_0.14.0            fansi_1.0.5
## [49] spatstat.sparse_3.0-3       httr_1.4.7
## [51] polyclip_1.10-6            abind_1.4-5
```

```
## [ 53] compiler_4.3.2           withr_2.5.2
## [ 55] fastDummies_1.7.3        highr_0.10
## [ 57] MASS_7.3-60              DelayedArray_0.28.0
## [ 59] tools_4.3.2              lmtest_0.9-40
## [ 61] zip_2.3.0                httpuv_1.6.12
## [ 63] future.apply_1.11.0      goftest_1.2-3
## [ 65] glmGamPoi_1.14.0        glue_1.6.2
## [ 67] nlme_3.1-164             promises_1.2.1
## [ 69] grid_4.3.2               Rtsne_0.16
## [ 71] cluster_2.1.6            reshape2_1.4.4
## [ 73] generics_0.1.3            gtable_0.3.4
## [ 75] spatstat.data_3.0-3     tidyR_1.3.0
## [ 77] data.table_1.14.8        XVector_0.42.0
## [ 79] utf8_1.2.4               BiocGenerics_0.48.1
## [ 81] spatstat.geom_3.2-7      RcppAnnoy_0.0.21
## [ 83] ggrepel_0.9.4             RANN_2.6.1
## [ 85] pillar_1.9.0              stringr_1.5.1
## [ 87] spam_2.10-0               RcppHNSW_0.5.0
## [ 89] later_1.3.1              rJava_1.0-10
## [ 91] splines_4.3.2             lattice_0.21-9
## [ 93] survival_3.5-7            deldir_2.0-2
## [ 95] tidyselect_1.2.0          miniUI_0.1.1.1
## [ 97] pbapply_1.7-2             knitr_1.45
## [ 99] gridExtra_2.3             IRanges_2.36.0
## [101] SummarizedExperiment_1.32.0 scattermore_1.2
## [103] stats4_4.3.2               xfun_0.41
## [105] Biobase_2.62.0             matrixStats_1.1.0
## [107] stringi_1.8.2              lazyeval_0.2.2
## [109] yaml_2.3.7                evaluate_0.23
## [111] codetools_0.2-19            xlsxjars_0.6.1
## [113] tibble_3.2.1               cli_3.6.1
## [115] uwot_0.1.16                xtable_1.8-4
## [117] reticulate_1.34.0            munsell_0.5.0
## [119] jquerylib_0.1.4             GenomeInfoDb_1.38.1
## [121] Rcpp_1.0.11                 globals_0.16.2
## [123] spatstat.random_3.2-2      png_0.1-8
## [125] parallel_4.3.2              ellipsis_0.3.2
## [127] dotCall164_1.1-1            sparseMatrixStats_1.14.0
## [129] bitops_1.0-7                listenv_0.9.0
## [131] viridisLite_0.4.2            scales_1.3.0
## [133] ggridges_0.5.4              crayon_1.5.2
## [135] leiden_0.4.3.1              purrrr_1.0.2
## [137] rlang_1.1.2                 cowplot_1.1.1
```