

```
## -----
## install.packages('devtools')
## devtools::install_github('immunogenomics/presto')
## -----
## After installation of presto, Seurat will automatically use the more
## efficient implementation (no further action necessary).
## This message will be shown once per session
```

```
## Calculating cluster MZ B
```

```
## Calculating cluster Transitional B
```

```
## Calculating cluster Mix
```

```
## Calculating cluster B1
```

```
## Calculating cluster GC B
```

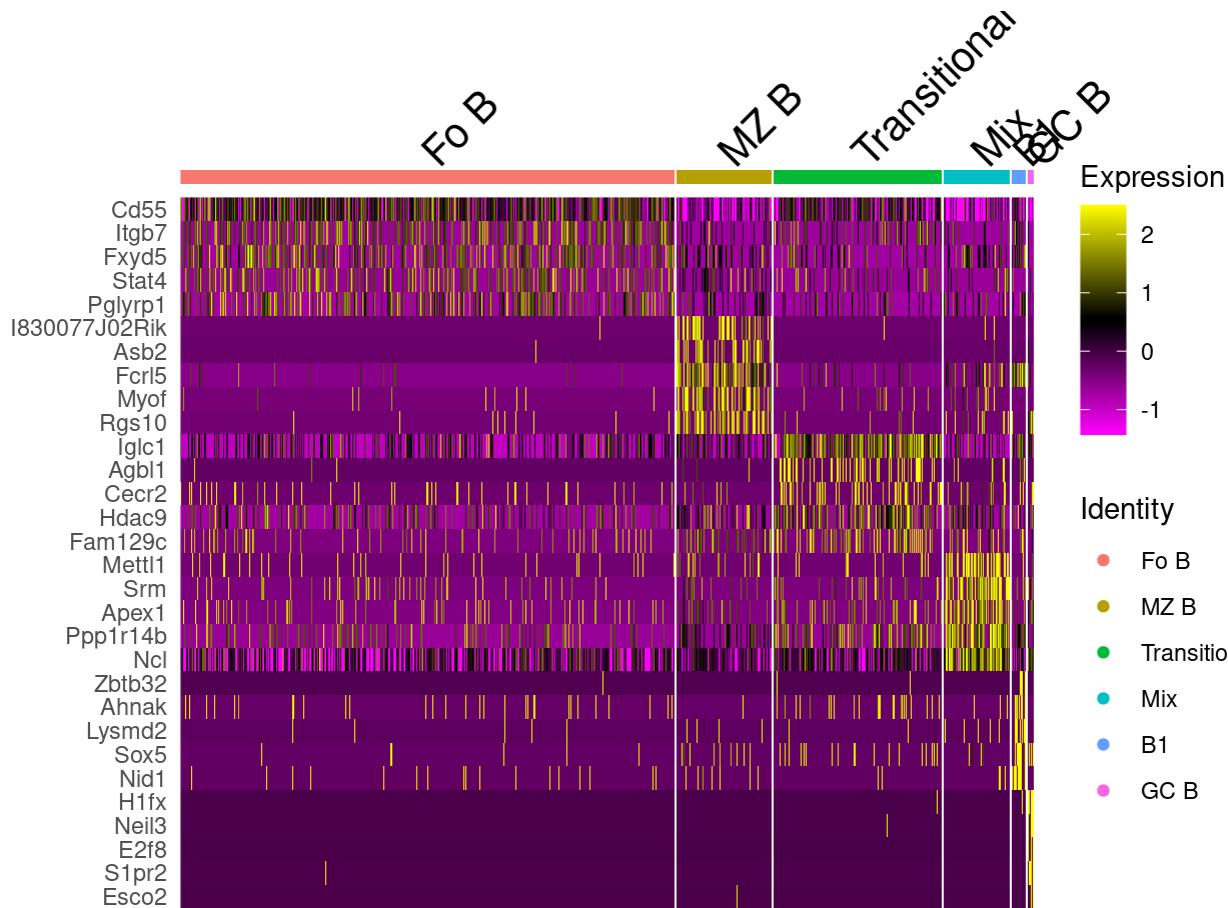
We print the two most relevant per cluster:

```
five_top_markers_SP <- all_markers_SP %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.25 | pct.2 > 0.25) & p_val_a
dj < 1e-20) %>%
  slice_min(n = 5, order_by = p_val_adj) %>%
  ungroup() -> top5_SP

five_top_markers_SP
```

```
## # A tibble: 30 × 7
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##      <dbl>     <dbl> <dbl> <dbl>    <dbl> <chr>
## 1 2.23e- 95     1.14  0.882  0.662  3.94e- 91 Fo B  Cd55
## 2 5.29e- 41     1.35  0.457  0.244  9.35e- 37 Fo B  Itgb7
## 3 8.96e- 35     1.28  0.462  0.282  1.58e- 30 Fo B  Fxyd5
## 4 1.14e- 27     1.32  0.387  0.233  2.01e- 23 Fo B  Stat4
## 5 4.59e- 27     1.25  0.408  0.257  8.10e- 23 Fo B  Pglyrp1
## 6 4.13e-176     5.65  0.375  0.008  7.30e-172 MZ B  I830077J02Rik
## 7 9.83e-170     5.97  0.346  0.005  1.74e-165 MZ B  Asb2
## 8 9.64e-162     3.26  0.67   0.09   1.70e-157 MZ B  Fcrl5
## 9 4.81e-161     3.96  0.522  0.044  8.50e-157 MZ B  Myof
## 10 1.65e-139    3.84  0.452  0.036  2.91e-135 MZ B  Rgs10
## # 20 more rows
```

```
DoHeatmap(Seurat_Object_SP_WT, features = top5_SP$gene)
```



```
average_expression <- AverageExpression(Seurat_Object_SP_WT, features = five_top_markers_SP$gene, return.seurat = TRUE)
```

```
## Warning: The following 30 features were not found in the HTO assay: Cd55, Itgb7, Fxyd5, Stat4, Pglyrp1, I830077J02Rik, Asb2, Fcrl5, Myof, Rgs10, Igcl1, Agbl1, Cecr2, Hdac9, Fam129c, Mettl1, Srm, Apex1, Ppp1r14b, Ncl, Zbtb32, Ahnak, Lysmd2, Sox5, Nid1, H1fx, Neil3, E2f8, S1pr2, Esco2
```

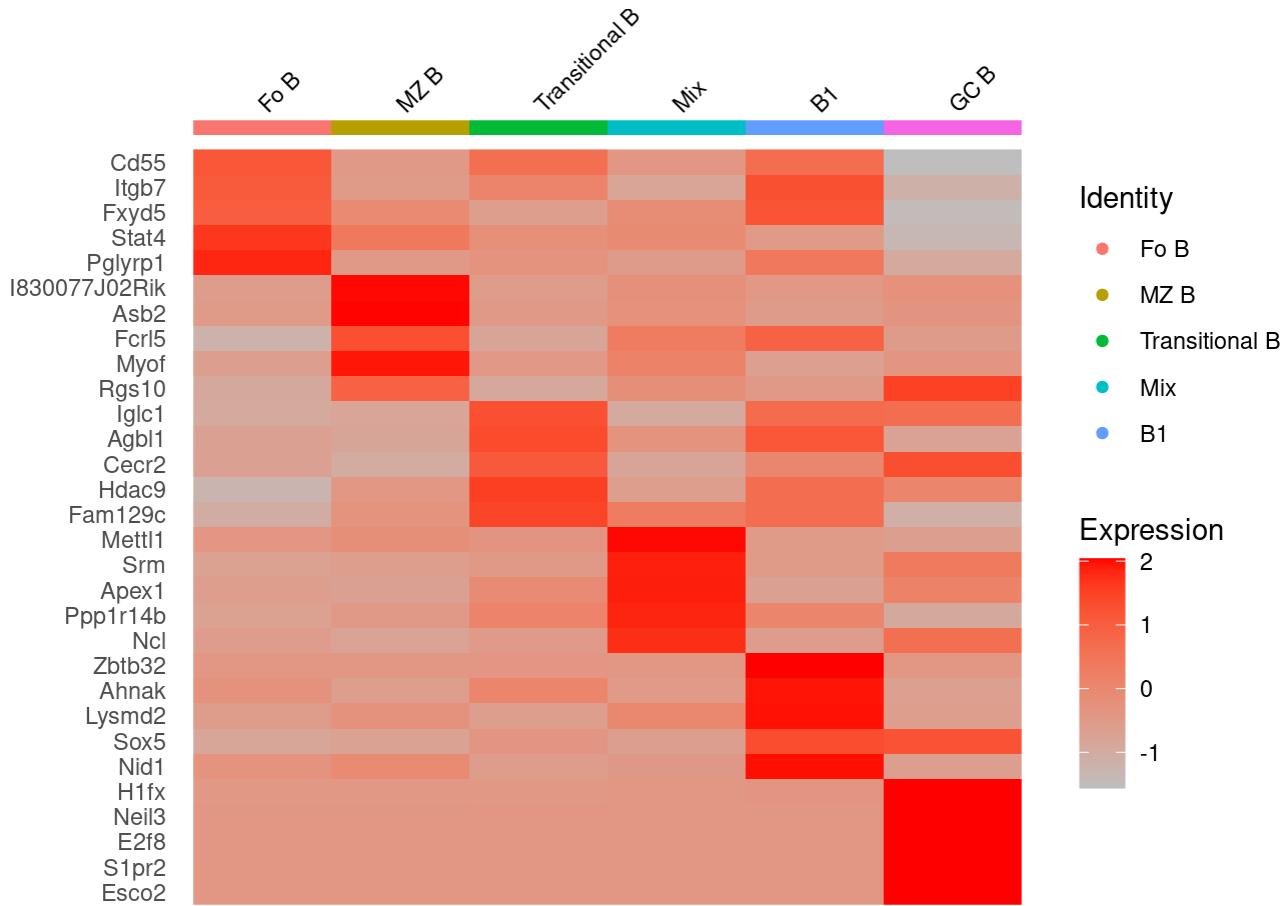
```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression, features = five_top_markers_SP$gene, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Wilcox WT Bone Marrow Analysis

```
all_markers_BM <- FindAllMarkers(object = Seurat_Object_BM_WT, only.pos = T, min.pct = 0.25, m
in.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Calculating cluster Mature B
```

```
## Calculating cluster Immature B
```

```
## Calculating cluster Small PreB
```

```
## Calculating cluster A12 unique
```

```
## Calculating cluster Pre-Pro B
```

```
## Calculating cluster Large PreB
```

```
## Calculating cluster Cycling Pre-ProB
```

```
## Calculating cluster Cycling ProB
```

```
## Calculating cluster ProB
```

```
## Calculating cluster Cycling Immature
```

```
## Calculating cluster Highly Mitochondrial
```

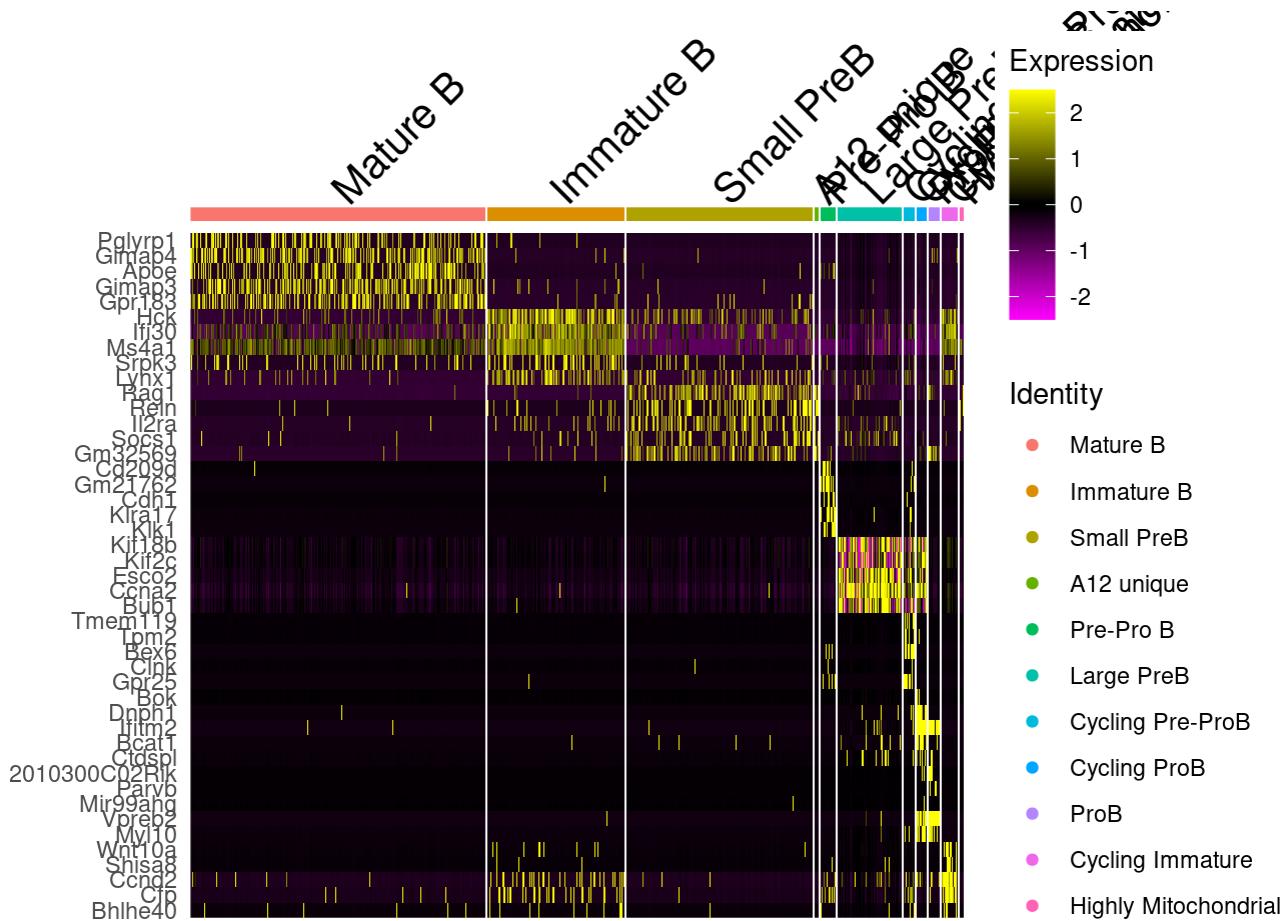
We print the five most relevant per cluster:

```
five_top_markers_BM <- all_markers_BM %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.3 | pct.2 > 0.3) & p_val_adj
< 1e-80) %>%
  slice_max(n = 5, order_by = avg_log2FC) %>%
  ungroup() -> top5_BM

five_top_markers_BM
```

```
## # A tibble: 45 × 7
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster   gene
##       <dbl>     <dbl>  <dbl> <dbl>    <dbl> <fct>   <chr>
## 1 1.31e-305     5.78 0.358 0.01 2.60e-301 Mature B Pglyrp1
## 2 0             5.76 0.464 0.018 0   Mature B Gimap4
## 3 9.40e-296     5.09 0.369 0.019 1.87e-291 Mature B Apoe
## 4 0             4.93 0.492 0.02  0   Mature B Gimap3
## 5 0             4.07 0.473 0.041 0   Mature B Gpr183
## 6 2.84e-278     2.55 0.671 0.181 5.64e-274 Immature B Hck
## 7 1.37e-282     2.53 0.849 0.439 2.73e-278 Immature B Ifi30
## 8 0             2.15 0.941 0.517 0   Immature B Ms4a1
## 9 1.25e- 85     1.90 0.341 0.112 2.48e- 81 Immature B Srpk3
## 10 1.17e-101    1.79 0.434 0.159 2.32e- 97 Immature B Lynx1
## # 35 more rows
```

```
DoHeatmap(Seurat_Object_BM_WT, features = top5_BM$gene)
```



```
average_expression <- AverageExpression(Seurat_Object_BM_WT, features = five_top_markers_BM$gene, return.seurat = TRUE)
```

```
## Warning: The following 45 features were not found in the HTO assay: Pglyrp1, Gimap4, Apoe, Gimap3, Gpr183, Hck, Ifi30, Ms4a1, Srpk3, Lynx1, Rag1, Reln, Il2ra, Socs1, Gm32569, Cd209d, Gm21762, Cdh1, Klra17, Klk1, Kif18b, Kif2c, Esco2, Ccna2, Bub1, Tmem119, Tpm2, Bex6, Clnk, Gpr25, Bok, Dnph1, Ifitm2, Bcat1, Ctdspl, 2010300C02Rik, Parvb, Mir99ahg, Vpreb2, Myl10, Wnt10a, Shisa8, Ccnd2, Cfp, Blhle40
```

```
## Warning: None of the features specified were found in the HTO assay.
```

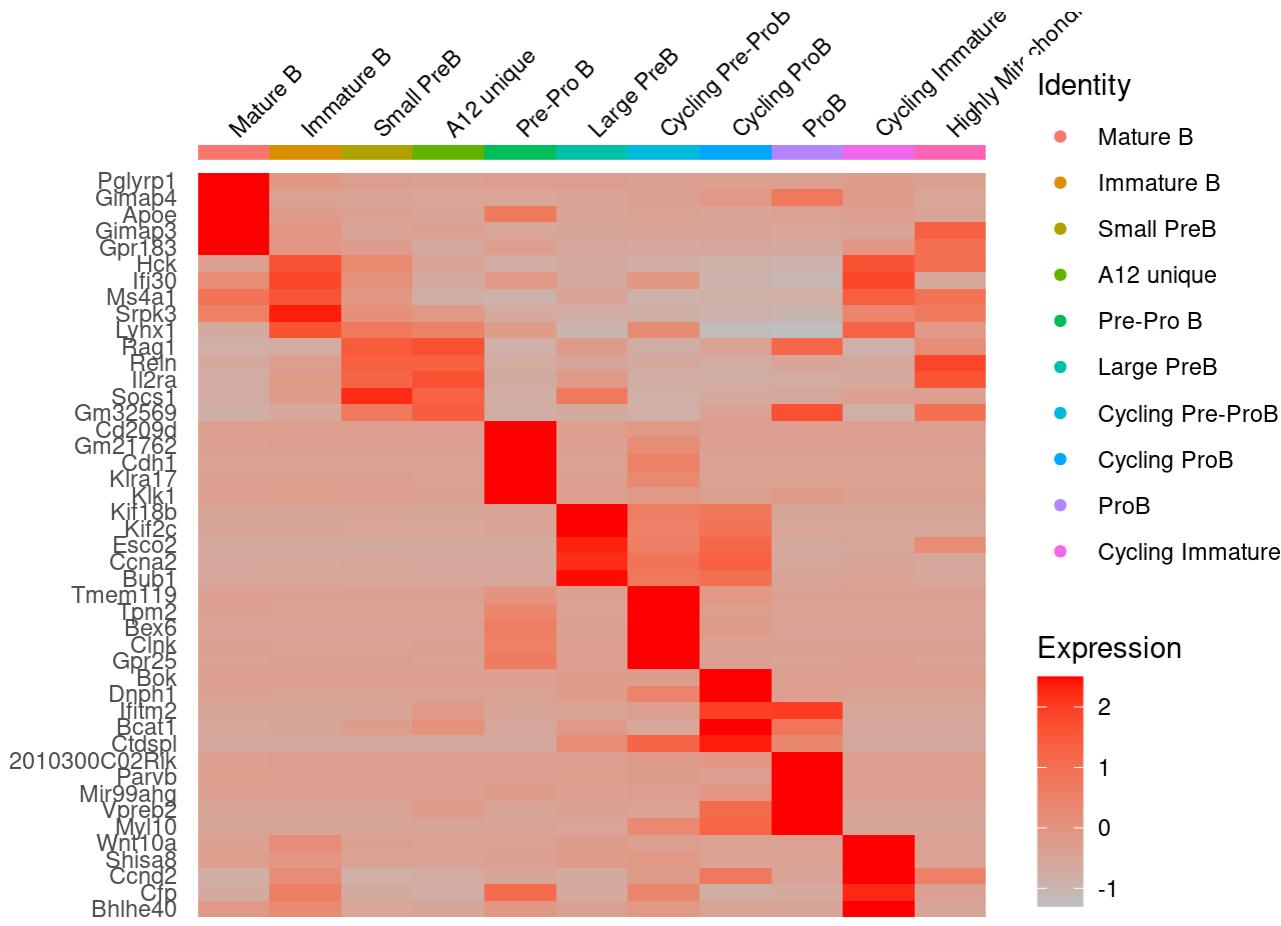
```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression, features = five_top_markers_BM$gene, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 45 rows containing missing values or values outside the scale range
```

```
## `geom_point()`).
```

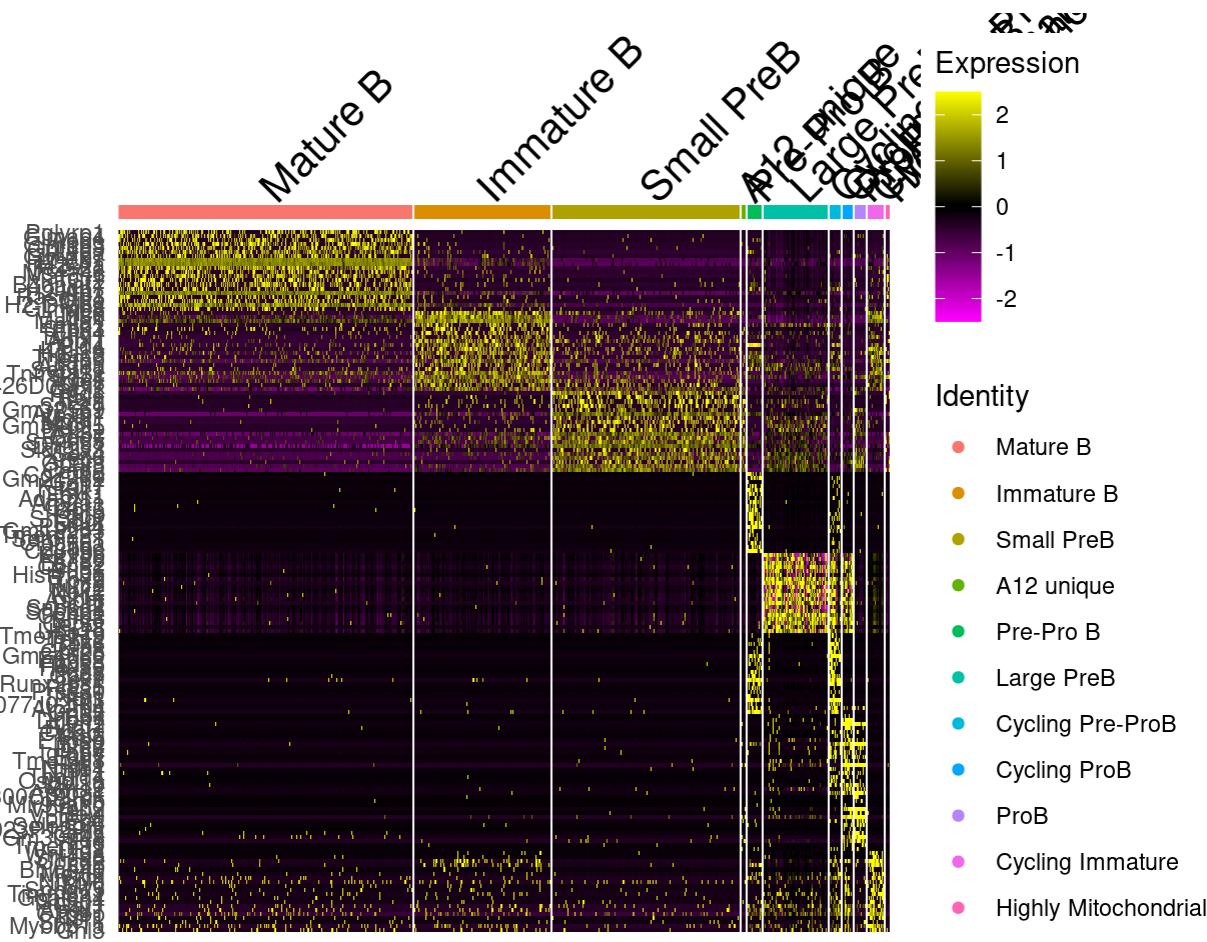


```
twenty_top_markers_BM <- all_markers_BM %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.3 | pct.2 > 0.3) & p_val_adj
< 1e-30) %>%
  slice_max(n = 20, order_by = avg_log2FC) %>%
  ungroup() -> top20_BM
```

```
twenty_top_markers_BM
```

```
## # A tibble: 180 × 7
##   p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##   <dbl>     <dbl> <dbl> <dbl>    <dbl> <fct> <chr>
## 1 1.31e-305     5.78 0.358 0.01 2.60e-301 Mature B Pglyrp1
## 2 0          5.76 0.464 0.018 0      Mature B Gimap4
## 3 9.40e-296     5.09 0.369 0.019 1.87e-291 Mature B Apoe
## 4 0          4.93 0.492 0.02  0      Mature B Gimap3
## 5 0          4.07 0.473 0.041 0      Mature B Gpr183
## 6 0          3.99 0.64  0.071 0      Mature B Ly6a
## 7 0          3.95 0.446 0.043 0      Mature B Gimap7
## 8 0          3.92 0.998 0.254 0      Mature B H2-Eb1
## 9 0          3.84 0.998 0.334 0      Mature B H2-Aa
## 10 0         3.84 0.681 0.062 0      Mature B Fcer2a
## # ... with 170 more rows
```

```
DoHeatmap(Seurat_Object_BM_WT, features = top20_BM$gene)
```



```
# Calcular la expresión promedio por cluster para los genes de interés
average_expression <- AverageExpression(Seurat_Object_BM_WT, features = top20_BM$gene, return.seurat = TRUE)
```

```
## Warning: The following 174 features were not found in the HTO assay: Pglyrp1,
## Gimap4, Apoe, Gimap3, Gpr183, Ly6a, Gimap7, H2-Eb1, H2-Aa, Fcer2a, Ms4a4c,
## March1, Cr2, B4galnt1, Itgb7, H2-Ab1, Rasgrp3, Ciita, H2-DMb2, Gimap8, Hck,
## Ifi30, Ms4a1, Emid1, Srpk3, Ehd4, Lynx1, Agb11, Pld4, Klhl14, Hdac9, Them6,
## Ptprj, Arid3a, Sept11, Tnfrsf13c, Sdc4, Iglc1, 4930426D05Rik, Cd72, Rag1, Reln,
## Il2ra, Socsl, Gm32569, Atplb1, Uch11, Mgst1, Gm34095, Bcl2l1, Arl15c, Coq7,
## Serinc5, Dnajc7, Slc12a3, Sox4, Xrcc6, Gpam, Tifa, Cecr2, Cd209d, Gm21762,
## Cdh1, Klra17, Klk1, Foxr1, Adam11, Atp2a1, Cxcr3, Ccr5, Ldhb, Sh3bgr, Ddr1,
## Ppfia4, Gm12253, Tmem221, Serpinh1, Clec10a, Lag3, Cd300c, Kif18b, Kif2c,
## Esco2, Ccna2, Bub1, Hist1h3g, Tpx2, Rrm2, Nuf2, Nek2, Kif11, Aspm, Pclaf,
## Sapcd2, Shcbp1, Ccnb2, Birc5, Cenpf, E2f8, Ube2c, Tmem119, Tpm2, Bex6, Clnk,
## Gpr25, Gm34680, Ephb2, Fads3, Hoxa9, Itgax, Chdh, Tex2, Runx2os1, Pdia5,
## Prss30, Nrp1, Flt3, I830077J02Rik, Atp2b4, Gria3, Bok, Dnph1, Ifitm2, Bcat1,
## Ctdspl, Lrp5, Enpep, Dntt, Drc7, Igfbp4, Egf17, Tmem98, Igll1, Nme4, Ttll11,
## Tbc1d4, Osbpl1a, Myl10, Gfra2, Arpp21, 2010300C02Rik, Parvb, Mir99ahg, Vpreb2,
## Heyl, Vpreb1, Eng, Selenom, A630023P12Rik, Grb7, Gm30948, Erg, Tmem91, Dlg2,
## Wnt10a, Shisa8, Ccnd2, Cfp, Blhhe40, Mettl1, Srm, Nfkbid, Myc, Snhg15, Timm8a1,
## Gpatch4, Snhg4, Gar1, Apex1, Clqbp, Tsrl, Snx11, Mybbpla, Gnl3
```

```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

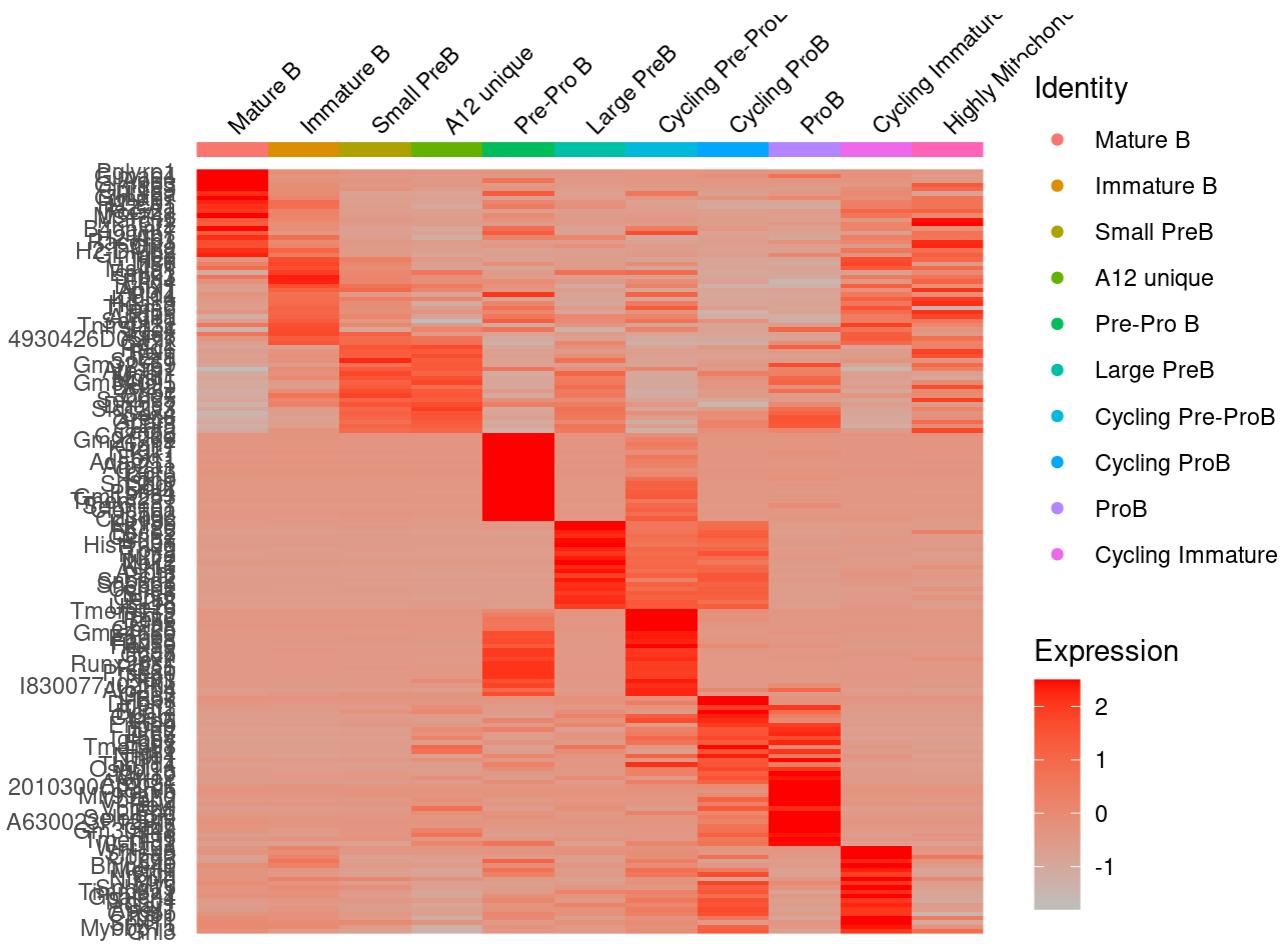
Crear el heatmap con los datos de expresión promedio

```
DoHeatmap(average_expression, features = top20_BM$gene, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
```

```
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 174 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Subclustering B1

Redo Scaling, PCA and clustering for B1

Identificacuon of highly variable features of B1

```
Seurat_Object_B1_cells <- subset(Seurat_Object_SP_selected_Bcells_ids, seurat_clusters_new == "B1")

Seurat_Object_B1_cells <- FindVariableFeatures(Seurat_Object_B1_cells, selection.method = "vst", nfeatures = 3000)
```

```
## Finding variable features for layer counts
```

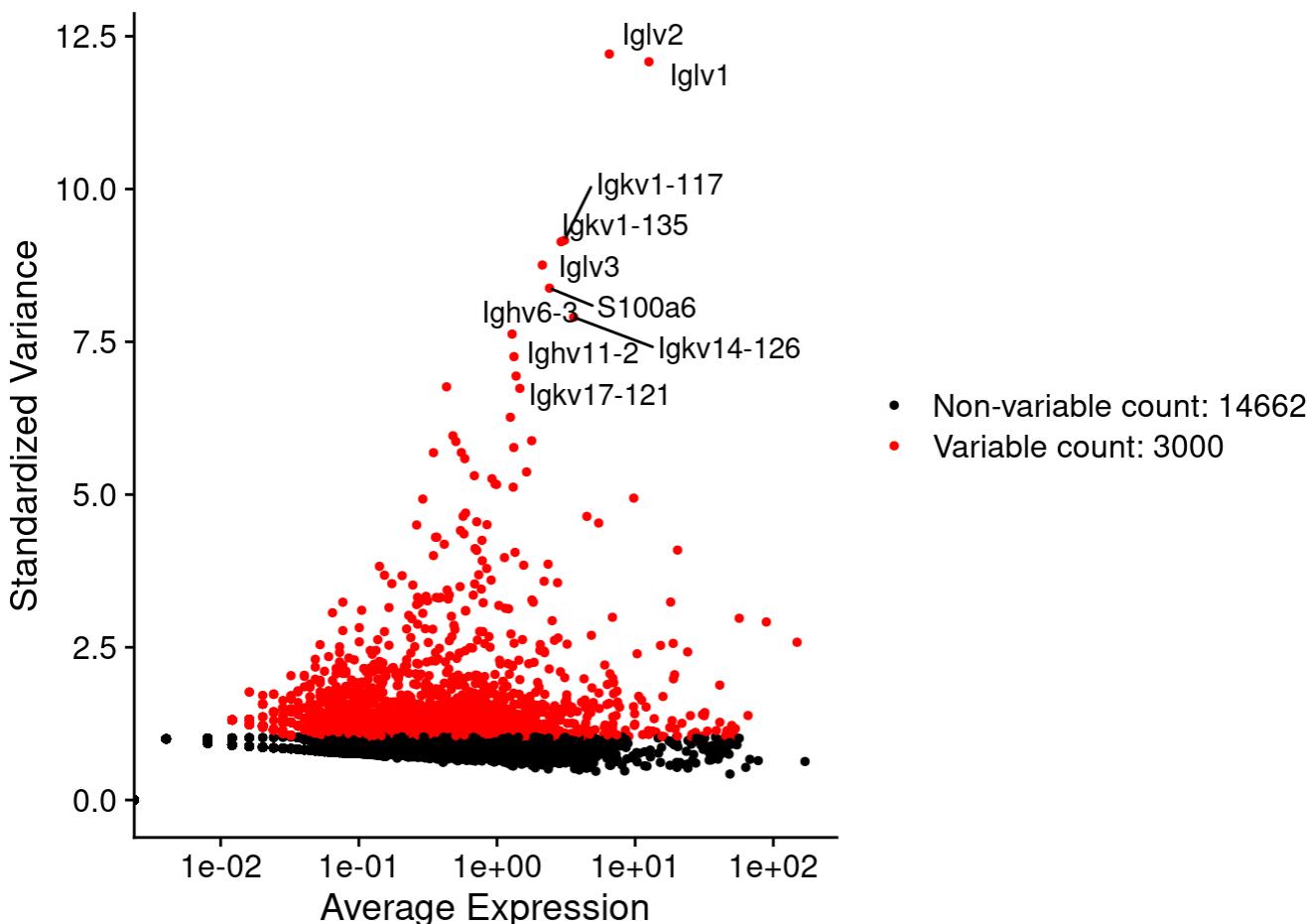
```
# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(Seurat_Object_B1_cells), 10)
```

```
# plot variable features with labels
plot1 <- VariableFeaturePlot(Seurat_Object_B1_cells)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
```

```
## When using repel, set xnudge and ynudge to 0 for optimal results
```

```
plot2
```

```
## Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```



```
variable_genes <- setdiff(VariableFeatures(Seurat_Object_B1_cells), c("A12"))
```

Scale B1 cells

```
Seurat_Object_B1_cells <- ScaleData(Seurat_Object_B1_cells, features = VariableFeatures(object
= Seurat_Object_B1_cells))

## Centering and scaling data matrix

## Warning: Different features in new layer data than already exists for
## scale.data
```

PCA of B1 cells

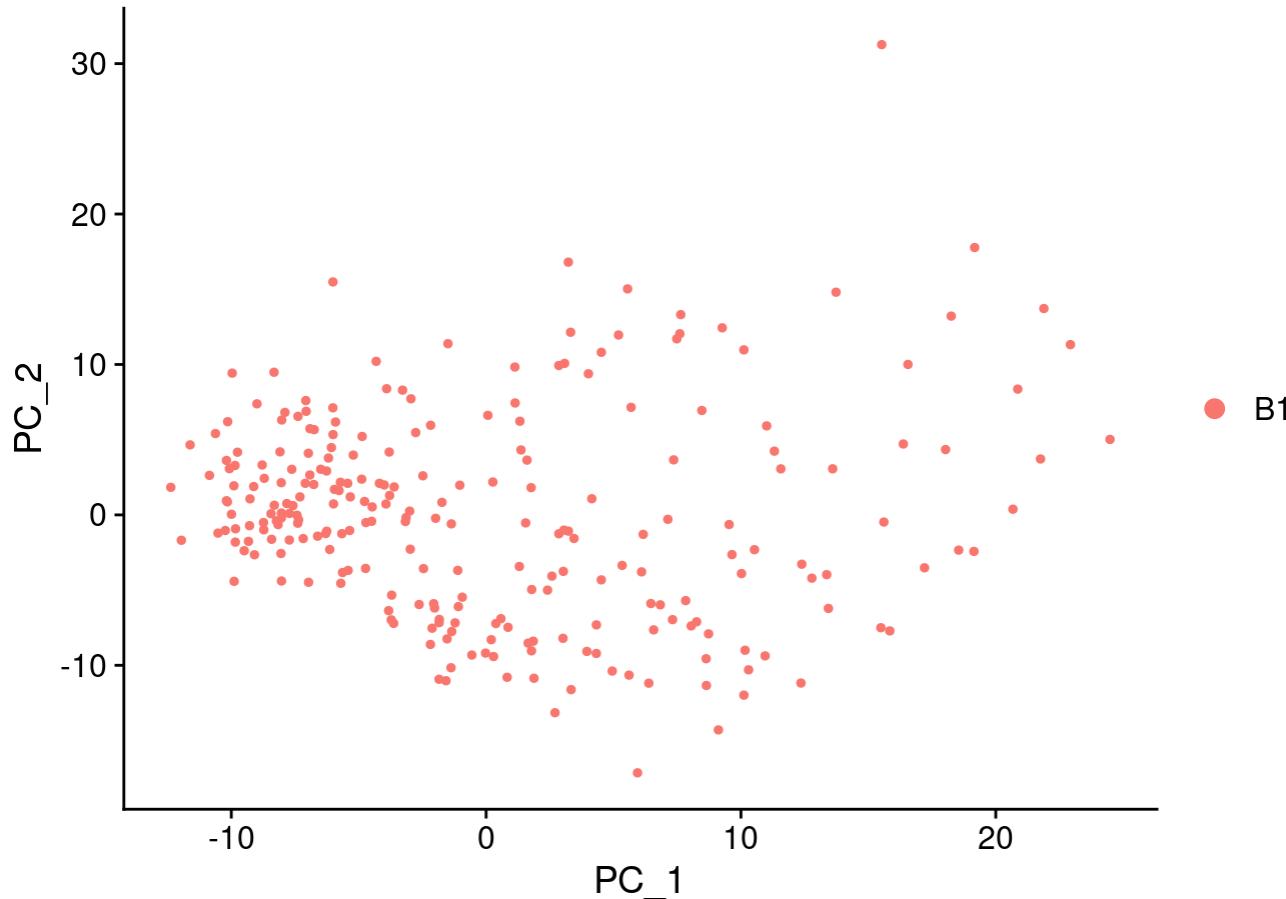
```
Seurat_Object_B1_cells <- RunPCA(Seurat_Object_B1_cells, features = variable_genes)

## PC_ 1
## Positive: Ft11, Actn1, Dnm3, Rftn1, Gm31718, Uap111, Hpgds, Atplb1, Zc3h12c, Litaf
## Mreg, Axl, Hivep3, Grn, Ube2a, Snx5, Fcer1g, Nos1, Mapk6, Sh3rf1
## Rilpl2, Abcb1b, Slc25a19, Plac8, Mef2a, Lztf1l, Mtdh, Npc2, Sqstm1, Mif
## Negative: Bank1, Ebfl1, Ralgps2, Gm31243, Cr2, Cyp4f18, Ptprj, Sell, Igkc, St6galnac3
## Rasgrp2, Coro2a, Cd55, Nedd9, Ighd, Acp5, Itpr3, Sh3pxd2a, Gm44734, Erollb
## Dennd4a, Pkib, Tsc22d3, Hck, Inpp4b, Il15ra, Cpm, Cd84, Il9r, Pgapl
## PC_ 2
## Positive: Srm, Eif4a1, Hsp90ab1, Nop58, Ncl, Nme1, Mett11, Eif5a, Hspd1, Ptma
## Nhp2, Fbl, Ppp1r14b, Ppia, Noc21, Hsp90aa1, Ccnd2, Cct3, Nop10, Set
## Psme2, Tubb4b, Hnrnpab, Gnl3, Inpp4b, 1810046K07Rik, Nolc1, Ranbp1, Kcnq5, Timm8a1
## Negative: Dnajc7, 2410006H16Rik, Dnm3, Plac8, Fcer1g, Apoe, Ig hg2c, Mzb1, Npc2, Rpl37
## Racgap1, Pafah1b3, Cdc25b, Gdi2, Rpl24, Trerf1, Pml, Smim24, Cmah, Gm45552
## Lockd, Scml4, Ciita, Rftn1, Ig hg3, Ass1, Ankub1, Selenop, Cpd, Gm13919
## PC_ 3
## Positive: Rps20, Rps26, Rpl37, Rpl41, Rps12, Rps18, Fcrl5, Rplp1, Rps27a, Rpl28
## Rps8, Rplp0, Rpsa, Rpl29, Arhgap24, Rpl10, Rpl36a, Rpl10a, Rgs10, Cd72
## Pold4, Dtx1, Rpl11, Eef1a1, Mzb1, Rpl3, Rpl7, Lipc, Myof, Eef1b2
## Negative: Vim, Tagln2, Klf2, Emp3, S100a10, Myadm, Crip1, Itgb7, Flna, Anxa2
## Ahnak, Rasa3, Lsp1, Rasgrp2, Myo1g, Gm15987, Fry, Gmfg, Gm45552, Fxyd5
## Tsc22d3, Jund, Arhgef18, Cd55, Gpx1, Tubala, Plec, Zbtb32, Lgals3, Ptp4a3
## PC_ 4
## Positive: Fcmr, Ephx1, Apoe, Ft11, Pgpep11, Laptm5, Pole4, Ig hd, Gstml1, Ptma
## Rpl7a, Cd69, Cxcr4, Bri3, S100a10, Fxyd5, Pcmtd1, Ccr7, Tsc22d3, Actn1
## Jund, Ank, Nab2, Snx5, Hsp90ab1, Cnot8, Rpl11, Retreg1, St6gall1, Ms4a4c
## Negative: Coro2a, Cd36, Rtn4rl2, Tenm4, Ptprj, 2700081015Rik, Ppp1r12b, Sema7a, Cd300lf, Cd1d1
## Gm26917, Wdr6, Cd274, Lipc, Tbxa2r, Pde4a, Eif2ak2, Kif5a, AY036118, Cep85
## Xist, Pld4, Fcrl5, Cacnale, Dnajc7, Glccil, Nedd4, Prr51, Gm42047, Slc29a3
## PC_ 5
## Positive: Gm50020, Dusp2, Akt3, Actn1, Cd72, Fcmr, Ccdc142os, Pcmtd1, Rtn4rl2, Zeb2
## Bach2, Nfatc1, Mfhas1, Neat1, Arhgap24, Osbpl18, Gm8369, Gm15327, Pitpnml, Zbtb38
## Pfkp, Fbxw10, Tmem163, Gm33843, Gramd1b, Myo5a, Gm42836, Ephx1, Tec, Man2a2
## Negative: Rpsa, Rpl14, Rps2, Rpl41, Rps27a, Rplp0, Eef1a1, Rps26, Rpl11, Rpl3
```

```
##      Rps8, Rpl7, Rpl29, Mki67, Btf3, Rpl28, Rpl7a, Sgo1, Rpl10a, Rps12
##      Eef1b2, Nme2, Cdca3, Aspm, Actg1, Rpl10, Rps20, Ccnb2, Rpl37, Eef2
```

Representation:

```
DimPlot(Seurat_Object_B1_cells, reduction = "pca")
```



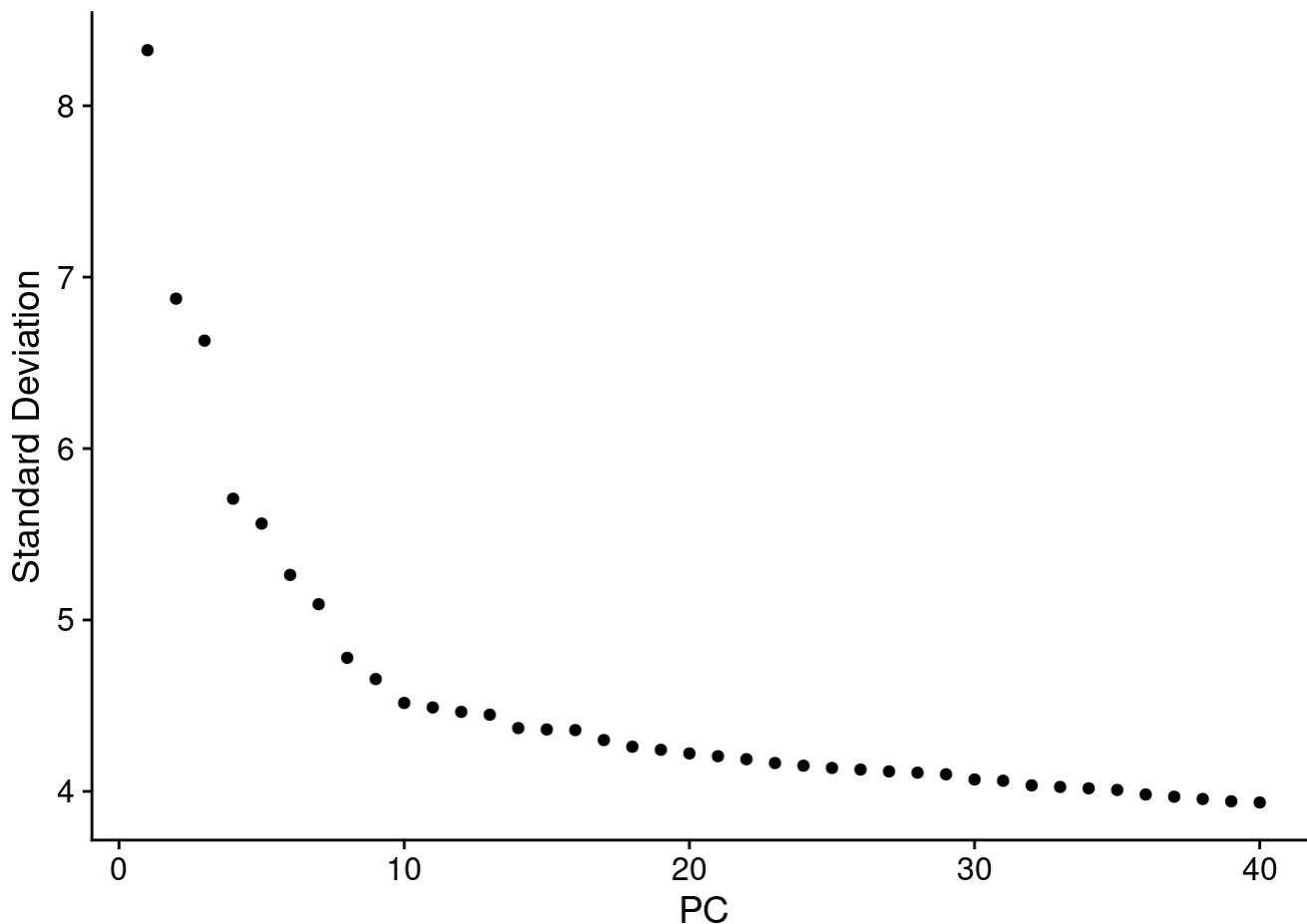
Determine the ‘dimensionality’ of the dataset

```
#Seurat_Object_B1_cells <- JackStraw(Seurat_Object_B1_cells, num.replicate = 100, dims = 40)
```

```
#Seurat_Object_B1_cells <- ScoreJackStraw(Seurat_Object_B1_cells, dims = 1:40)
```

```
#JackStrawPlot(Seurat_Object_B1_cells, dims = 1:40)
```

```
ElbowPlot(Seurat_Object_B1_cells, ndims=40)
```



B1 cells clusters

```
Seurat_Object_B1_cells <- FindNeighbors(Seurat_Object_B1_cells, dims = 1:10)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
Seurat_Object_B1_cells <- FindClusters(Seurat_Object_B1_cells, resolution = 0.3)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 248
## Number of edges: 8310
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7621
## Number of communities: 2
## Elapsed time: 0 seconds
```

```
Seurat_Object_B1_cells <- RunUMAP(Seurat_Object_B1_cells, dims = 1:10)
```

```
## 13:38:15 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 13:38:15 Read 248 rows and found 10 numeric columns
```

```
## 13:38:15 Using Annoy for neighbor search, n_neighbors = 30
```

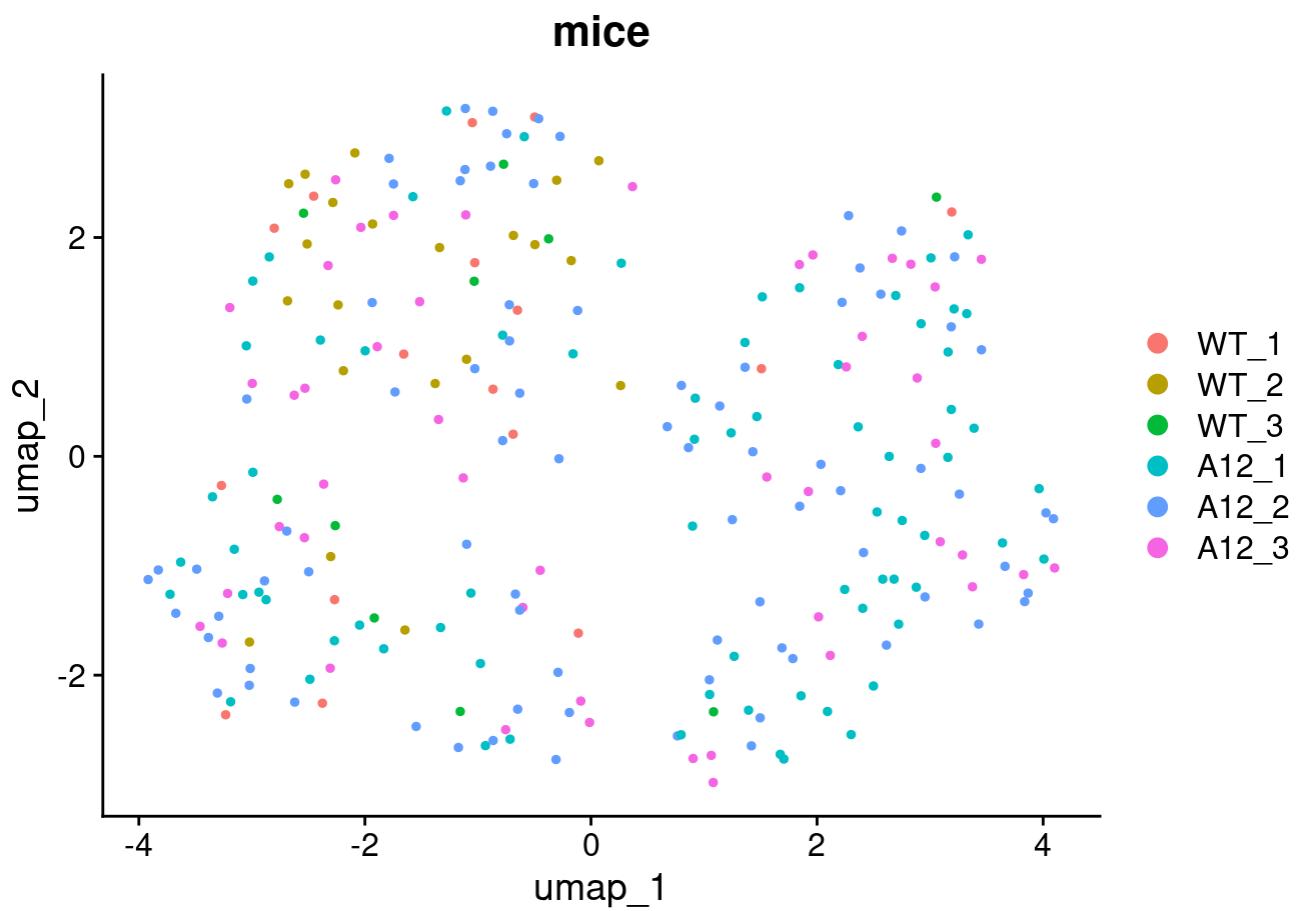
```
## 13:38:15 Building Annoy index with metric = cosine, n_trees = 50
```

```
## 0% 10 20 30 40 50 60 70 80 90 100%
```

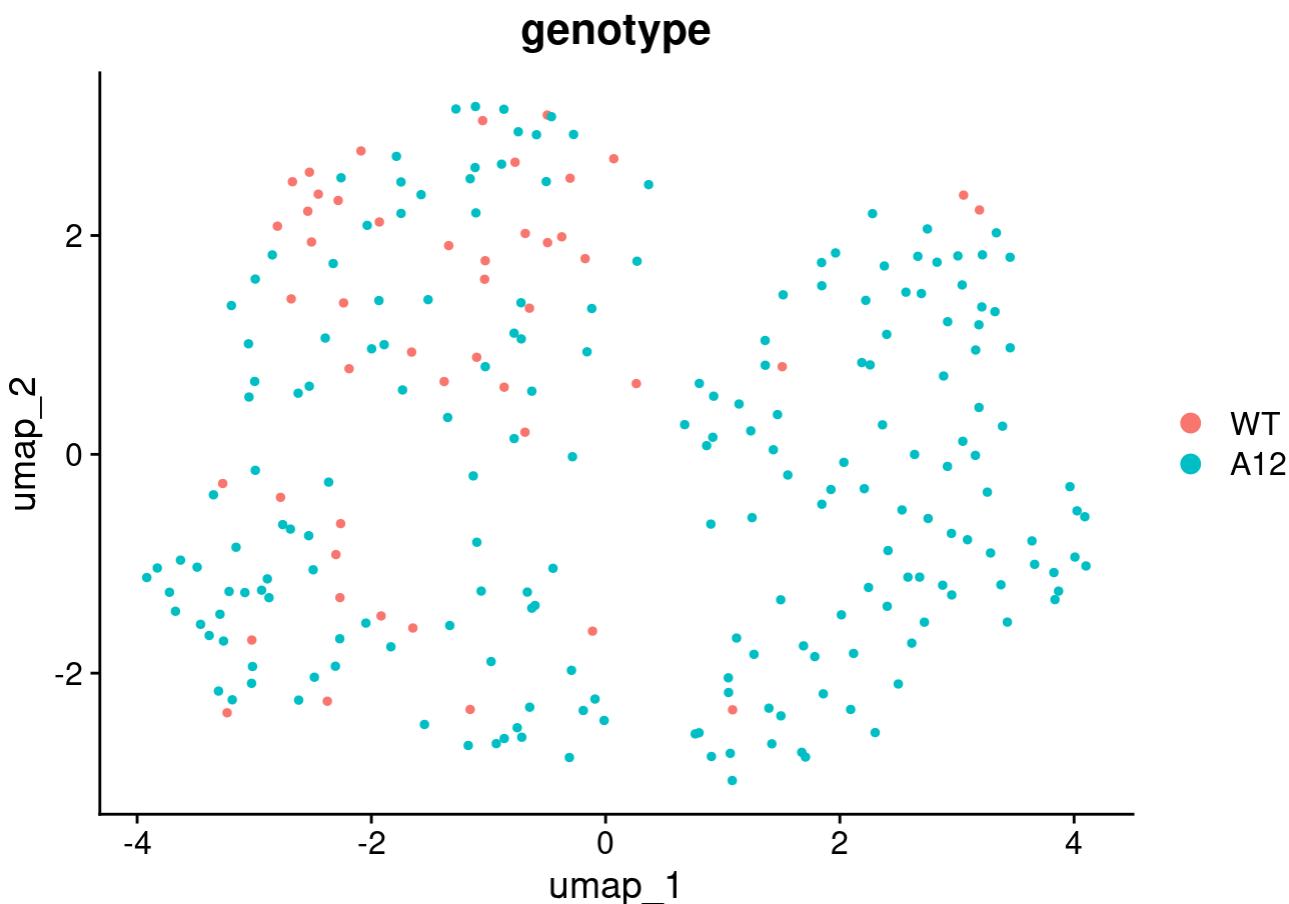
```
## [----|----|----|----|----|----|----|----|----|----|
```

```
## ****|  
## 13:38:15 Writing NN index file to temp file /tmp/RtmpZuvVCA/file12ff492ed966d  
## 13:38:15 Searching Annoy index using 1 thread, search_k = 3000  
## 13:38:15 Annoy recall = 100%  
## 13:38:16 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors  
= 30  
## 13:38:18 Initializing from normalized Laplacian + noise (using RSpectra)  
## 13:38:18 Commencing optimization for 500 epochs, with 8430 positive edges  
## 13:38:19 Optimization finished
```

```
DimPlot(Seurat_Object_B1_cells, reduction = "umap", group.by = "mice")
```

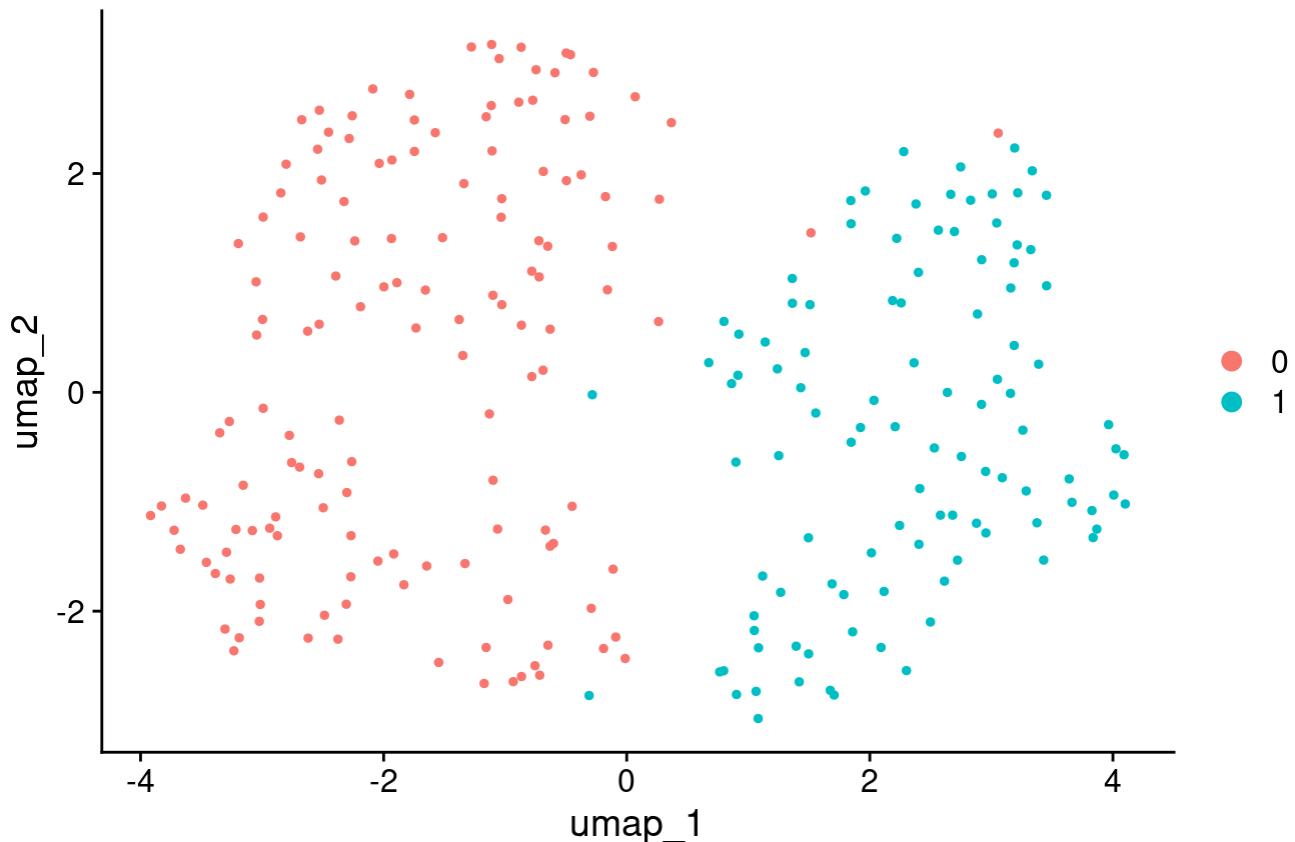


```
DimPlot(Seurat_Object_B1_cells, reduction = "umap", group.by = "genotype")
```

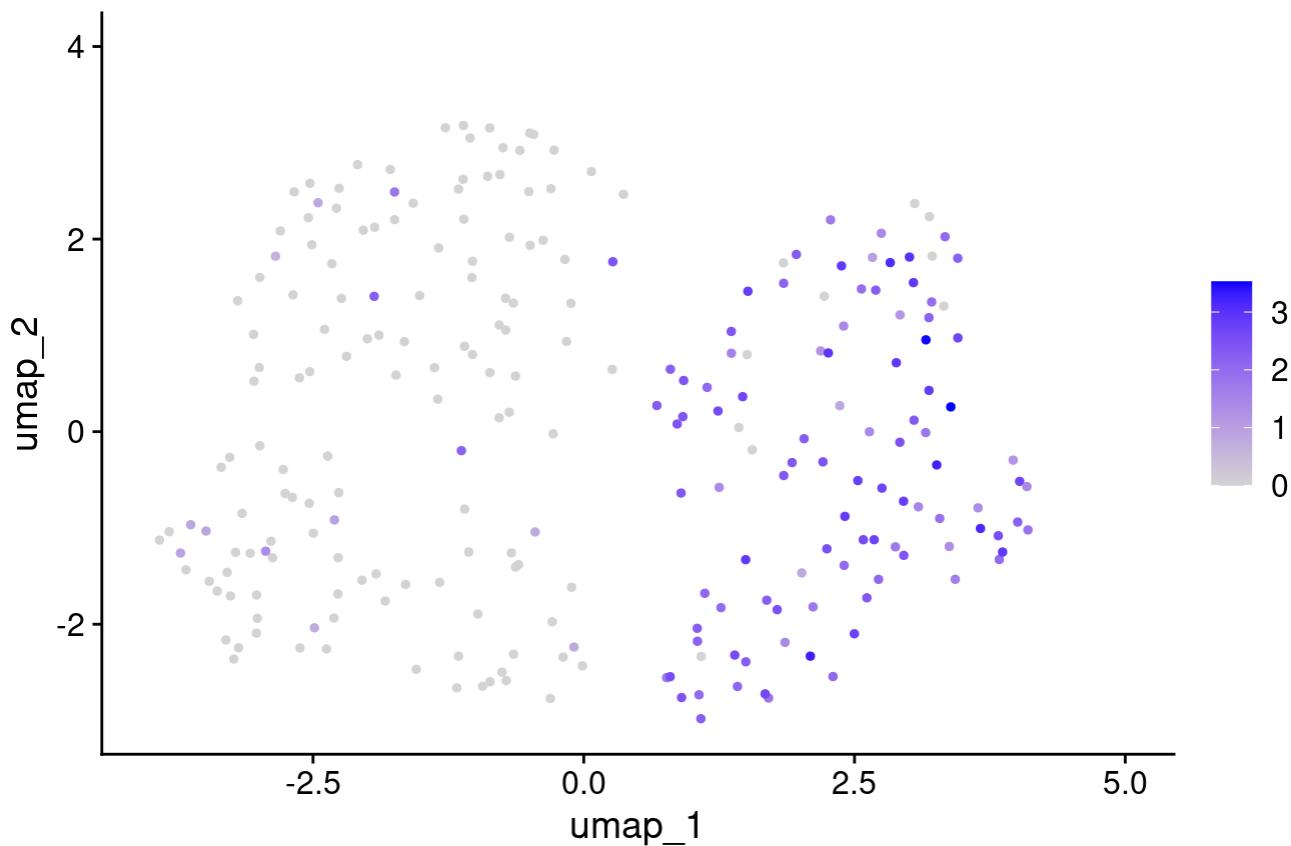


```
DimPlot(Seurat_Object_B1_cells, reduction = "umap", group.by = "seurat_clusters")
```

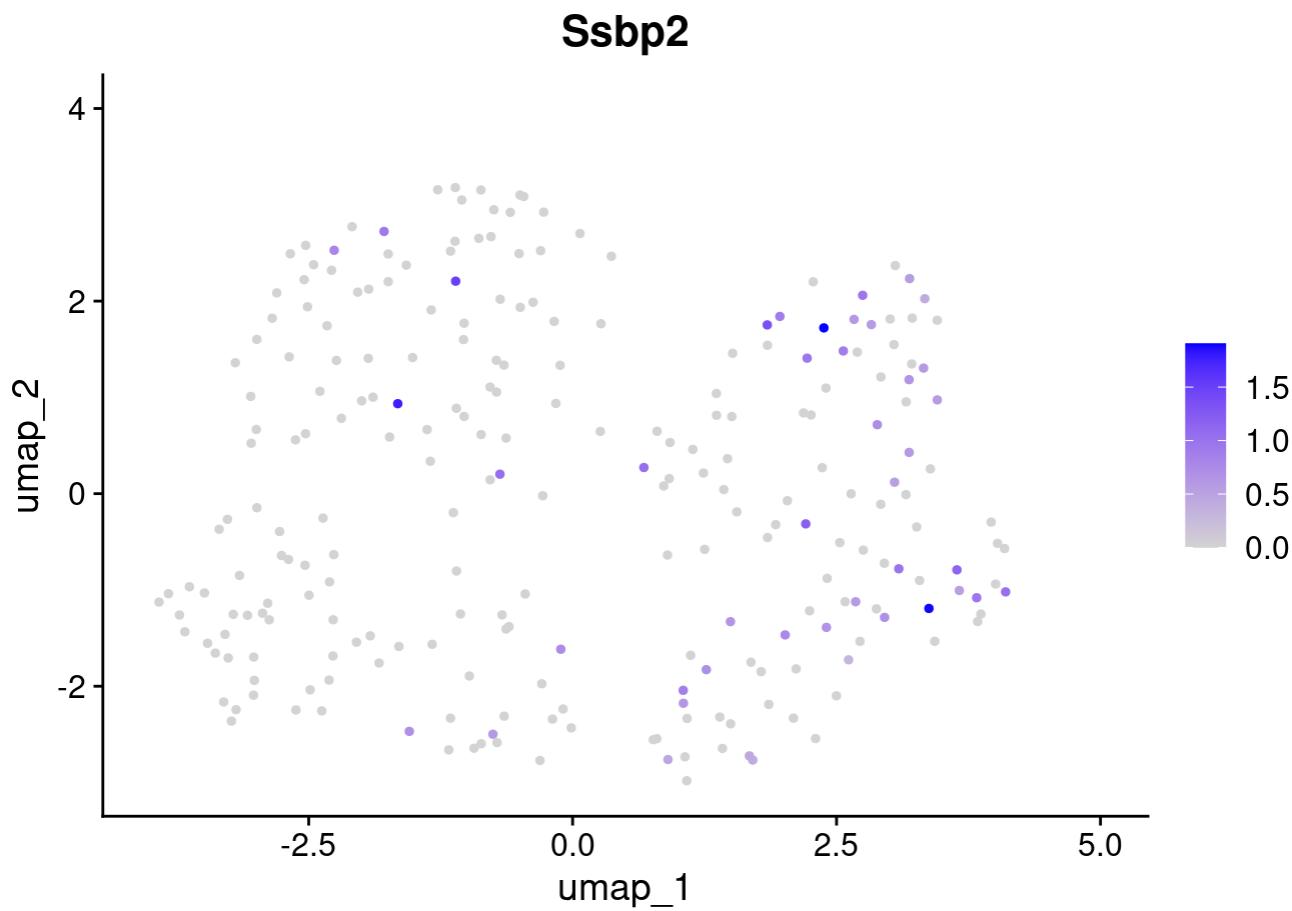
seurat_clusters



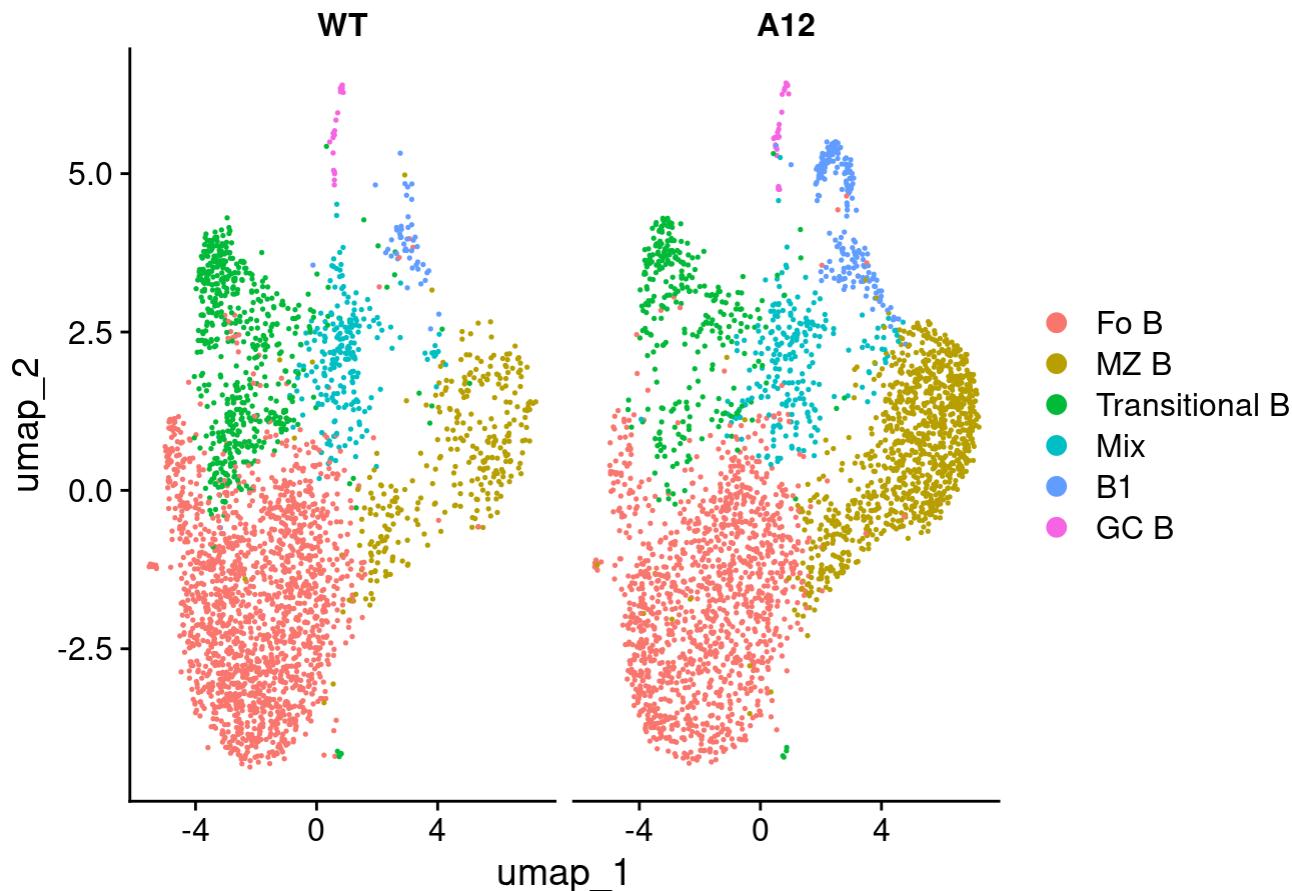
```
FeaturePlot(Seurat_Object_B1_cells, features = "A12")
```

A12

```
FeaturePlot(Seurat_Object_B1_cells, features = "Ssbp2")
```

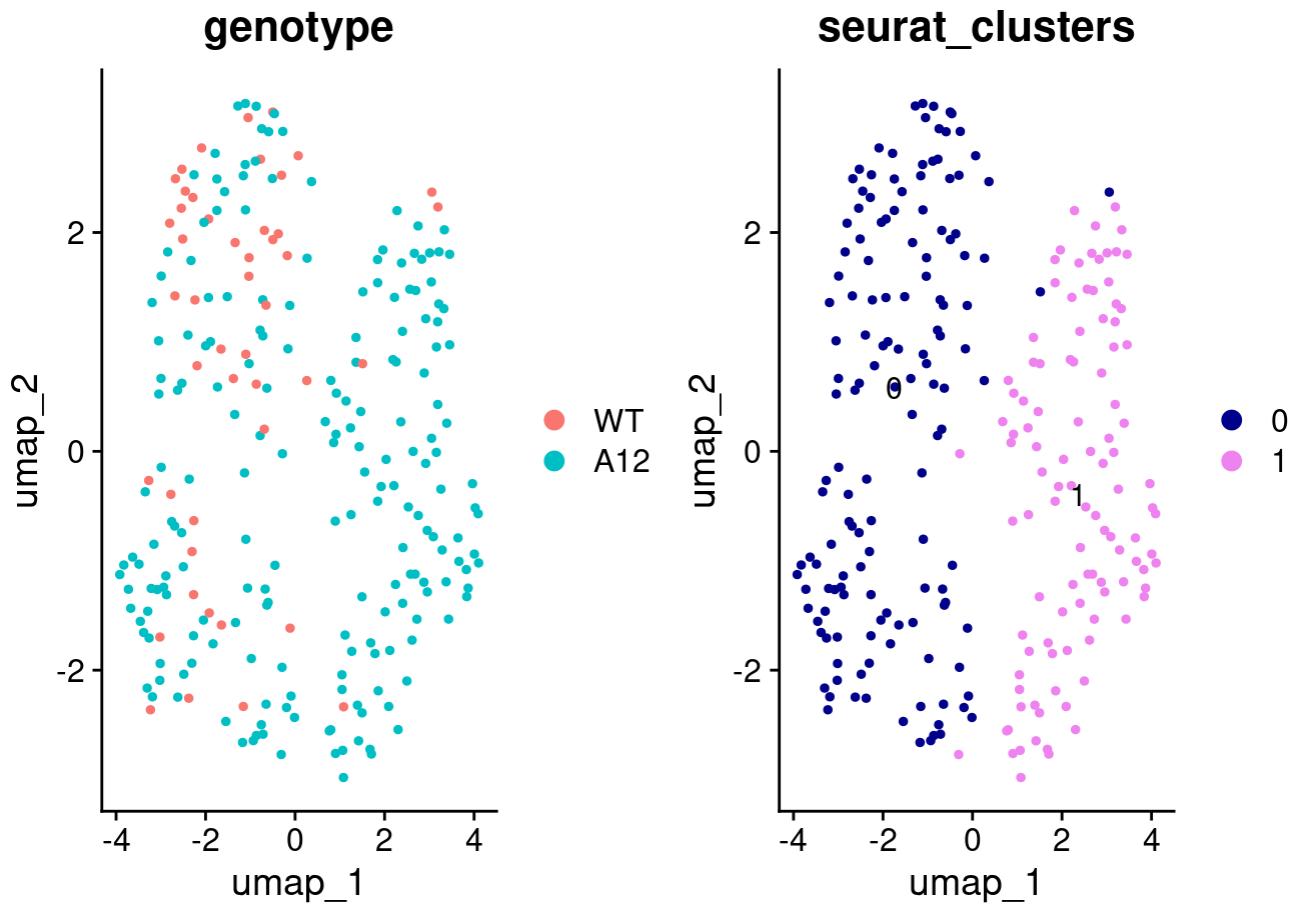


```
DimPlot(Seurat_Object_SP_selected_Bcells_idents, split.by = "genotype")
```



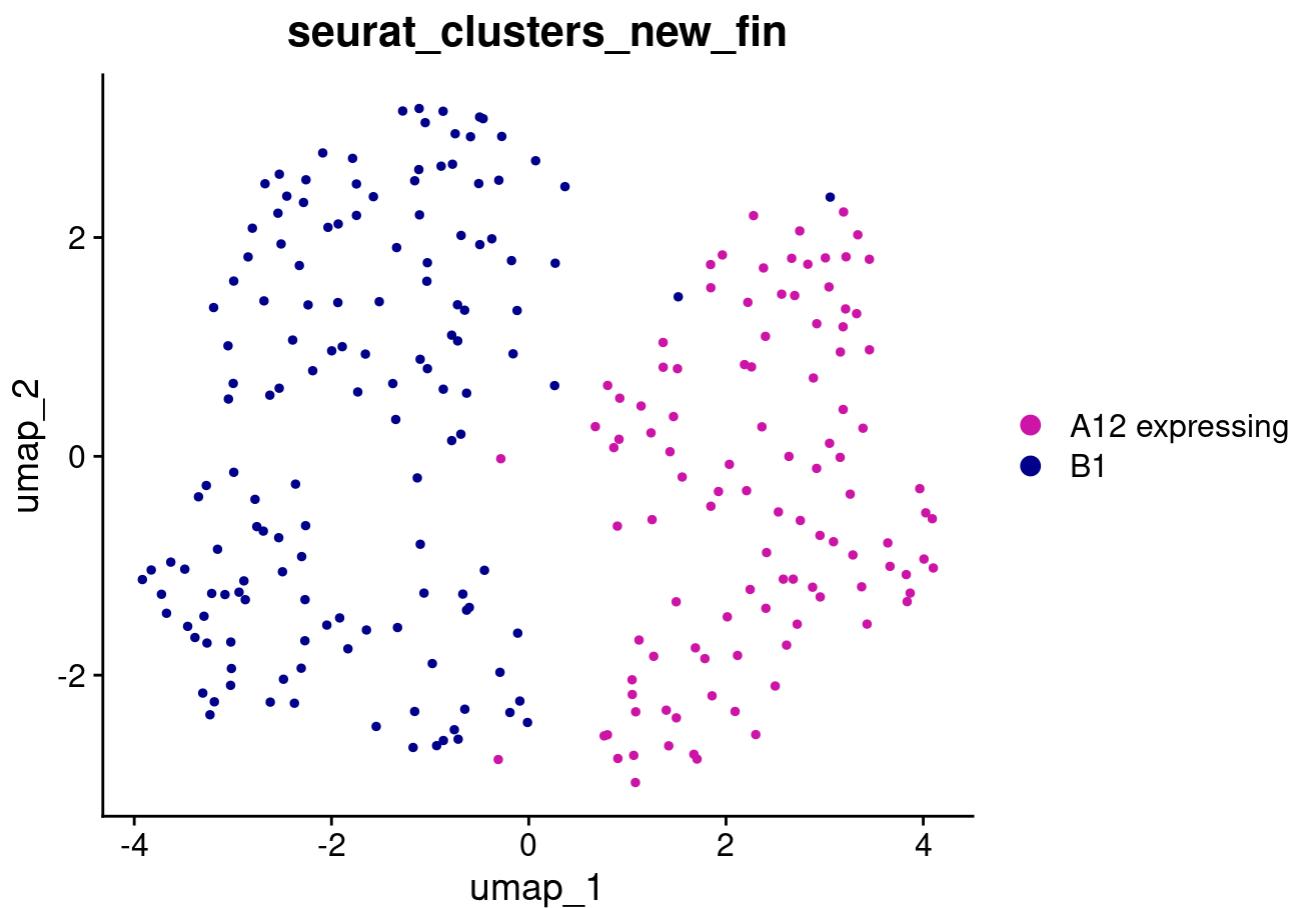
Rename clusters and analysis for SP

```
DimPlot(Seurat_Object_B1_cells, reduction = "umap", group.by = "genotype") + DimPlot(Seurat_Object_B1_cells, reduction = "umap", group.by = "seurat_clusters", label = TRUE, cols = c("darkblue", "violet"))
```



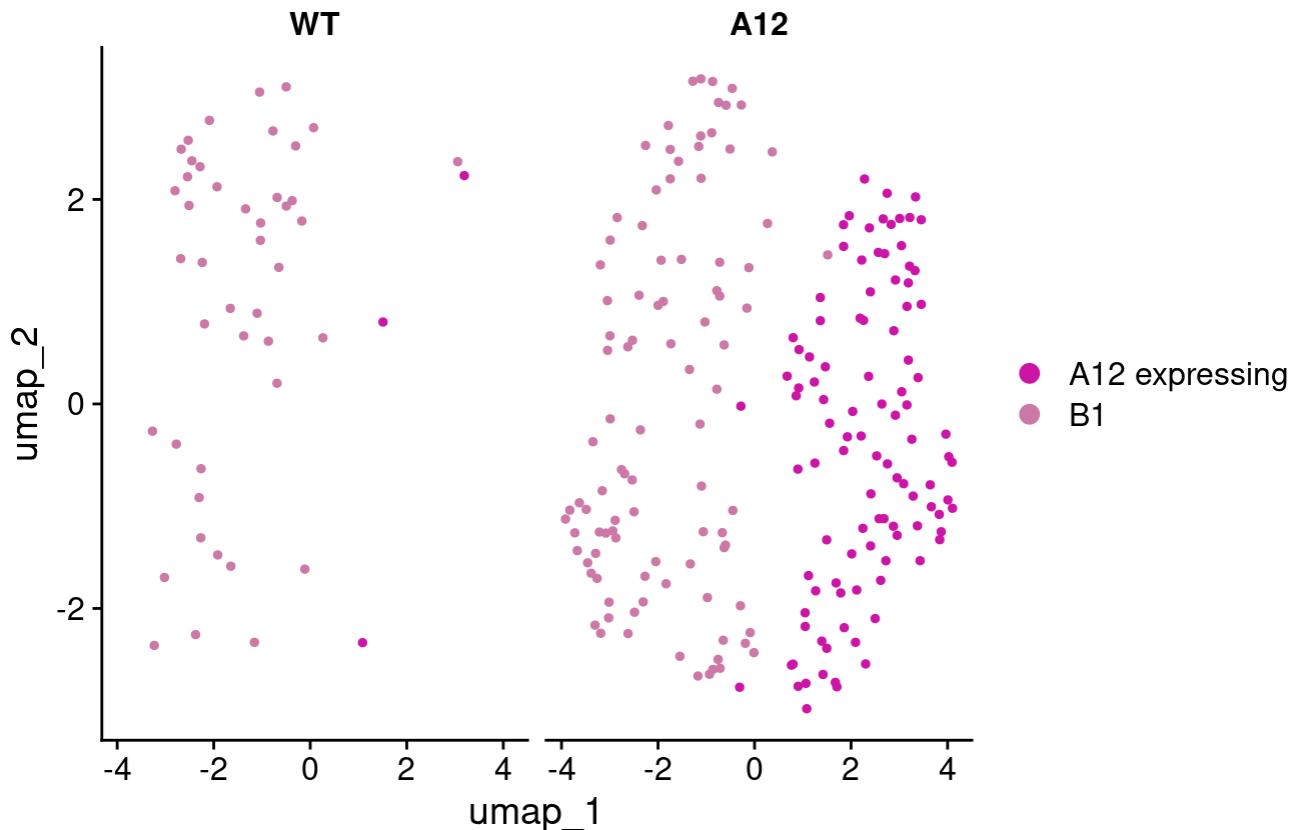
```
Seurat_Object_B1_cells_idents <- RenameIdents(Seurat_Object_B1_cells, "1" = "A12 expressing",
"0" = "B1")
Seurat_Object_B1_cells_idents$seurat_clusters_new_fin <- Idents(Seurat_Object_B1_cells_idents)
```

```
#plot the cells from Blimp using the transferred clustering
DimPlot(Seurat_Object_B1_cells_idents, reduction = "umap", cols= c("A12 expressing" = "#CC14A7",
"B1" = "darkblue"), group.by = "seurat_clusters_new_fin")
```



```
DimPlot(Seurat_Object_B1_cells_idents, reduction = "umap", cols= c("A12 expressing" = "#CC14A7", "B1" = "#CC79A7"), split.by = "genotype", group.by = "seurat_clusters_new_fin")
```

seurat_clusters_new_fin



```
all_markers_B1_subcl <- FindAllMarkers(object = Seurat_Object_B1_cells_idents, only.pos = T, m
in.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Calculating cluster A12 expressing
```

```
## Calculating cluster B1
```

We print the five most relevant per cluster:

```
five_top_markers_B1_subcl <- all_markers_B1_subcl %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.3 | pct.2 > 0.3) & p_val_adj
< 1e-4) %>%
  slice_max(n = 20, order_by = avg_log2FC) %>%
  ungroup()

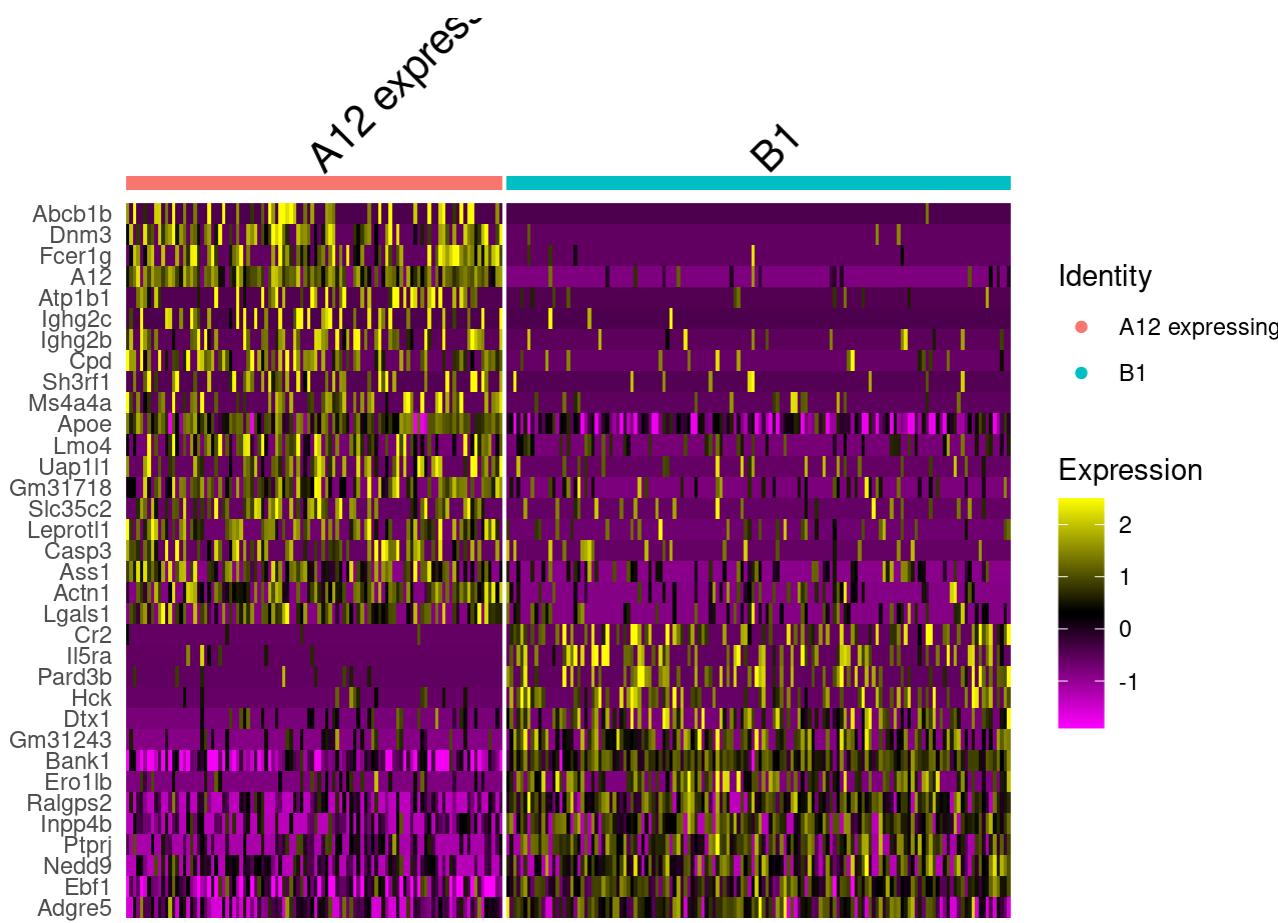
five_top_markers_B1_subcl
```

```
## # A tibble: 38 × 7
##   p_val avg_log2FC pct.1 pct.2 p_val_adj cluster   gene
##   <dbl>    <dbl> <dbl> <dbl>    <dbl> <fct>    <chr>
## 1 6.29e-14    5.28 0.358 0.007  1.11e-9 A12 expressing Abcb1b
## 2 2.17e-25    5.09 0.642 0.021  2.07e-21 A12 expressing Dnm3
## 3 6.57e-24    4.85 0.623 0.035  1.16e-19 A12 expressing Fcer1g
```

```
## 4 2.81e-36 4.32 0.896 0.106 4.96e-32 A12 expressing A12
## 5 2.82e- 9 3.90 0.368 0.077 4.98e- 5 A12 expressing Atp1b1
## 6 4.78e-11 3.52 0.33 0.021 8.45e- 7 A12 expressing Ig hg2c
## 7 2.43e-11 2.81 0.453 0.085 4.29e- 7 A12 expressing Ig hg2b
## 8 2.37e-15 2.60 0.575 0.099 4.18e-11 A12 expressing Cpd
## 9 4.65e- 9 2.47 0.368 0.063 8.22e- 5 A12 expressing Sh3rf1
## 10 6.31e-10 2.46 0.453 0.113 1.12e- 5 A12 expressing Ms4a4a
### # 28 more rows
```

```
DoHeatmap(Seurat_Object_B1_cells_idents, features = five_top_markers_B1_subcl$gene)
```

```
## Warning in DoHeatmap(Seurat_Object_B1_cells_idents, features =
## five_top_markers_B1_subcl$gene): The following features were omitted as they
## were not found in the scale.data slot for the RNA assay: Cxcr5, Pdcd4, Sipal,
## Scd1
```



```
average_expression <- AverageExpression(Seurat_Object_B1_cells_idents, features = five_top_markers_B1_subcl$gene, return.seurat = TRUE)
```

```
## Warning: The following 38 features were not found in the HTO assay: Abcb1b,
## Dnm3, Fcer1g, A12, Atp1b1, Ig hg2c, Ig hg2b, Cpd, Sh3rf1, Ms4a4a, Apoe, Lmo4,
## Uap1l1, Gm31718, Slc35c2, Leprotl1, Casp3, Ass1, Actn1, Lgals1, Cr2, Il5ra,
## Pard3b, Hck, Dtx1, Gm31243, Bank1, Ero1lb, Ralgps2, Inpp4b, Scd1, Ptprj, Sipal,
## Nedd9, Ebf1, Pdcd4, Cxcr5, Adgre5
```

```
## Warning: None of the features specified were found in the HTO assay.
```

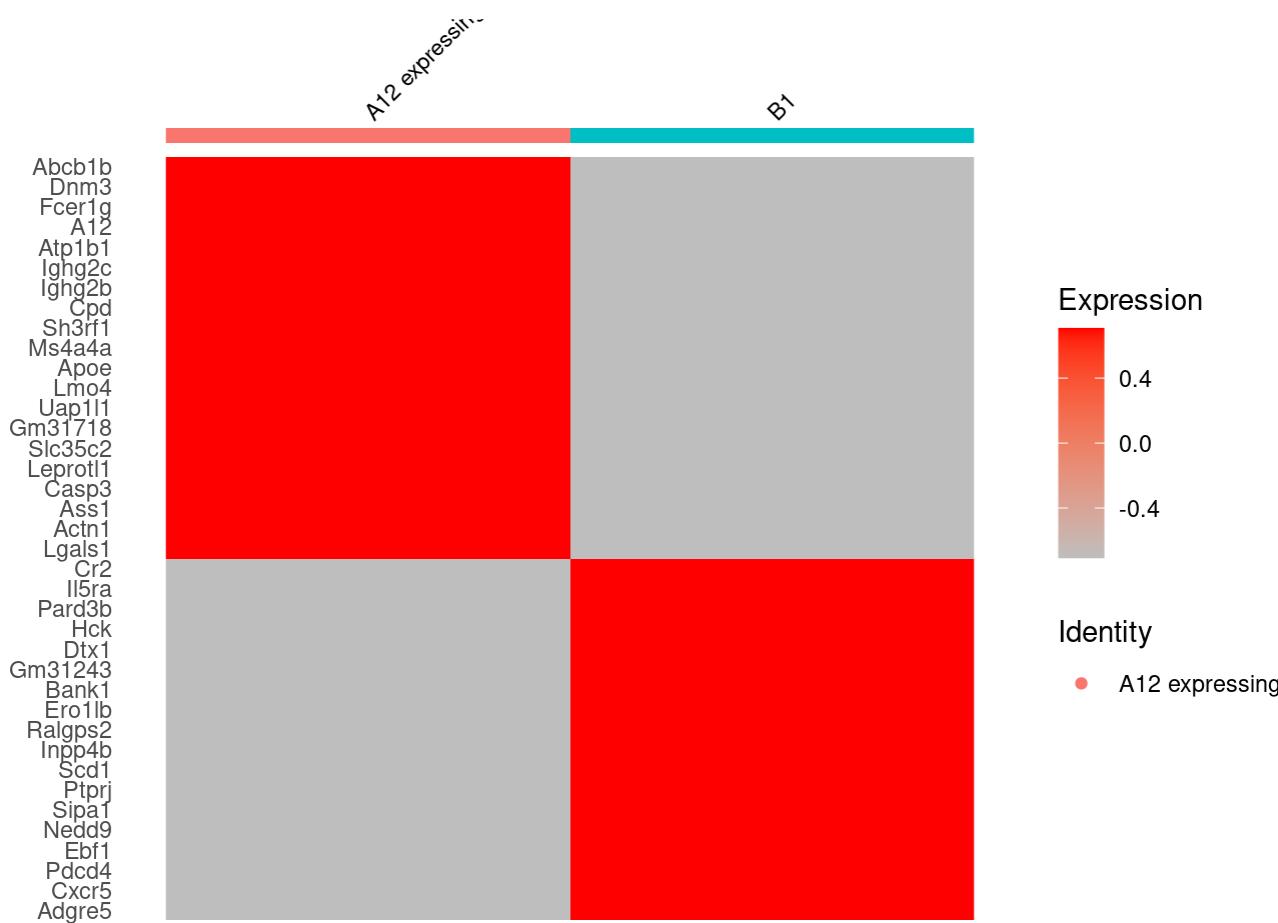
```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression, features = five_top_markers_B1_subcl$gene, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
```

```
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 38 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Rename in original clusters

```
A12_expressing_cells <- Cells(Seurat_Object_B1_cells_idents)[which(Seurat_Object_B1_cells_idents$seurat_clusters == "1")]
```

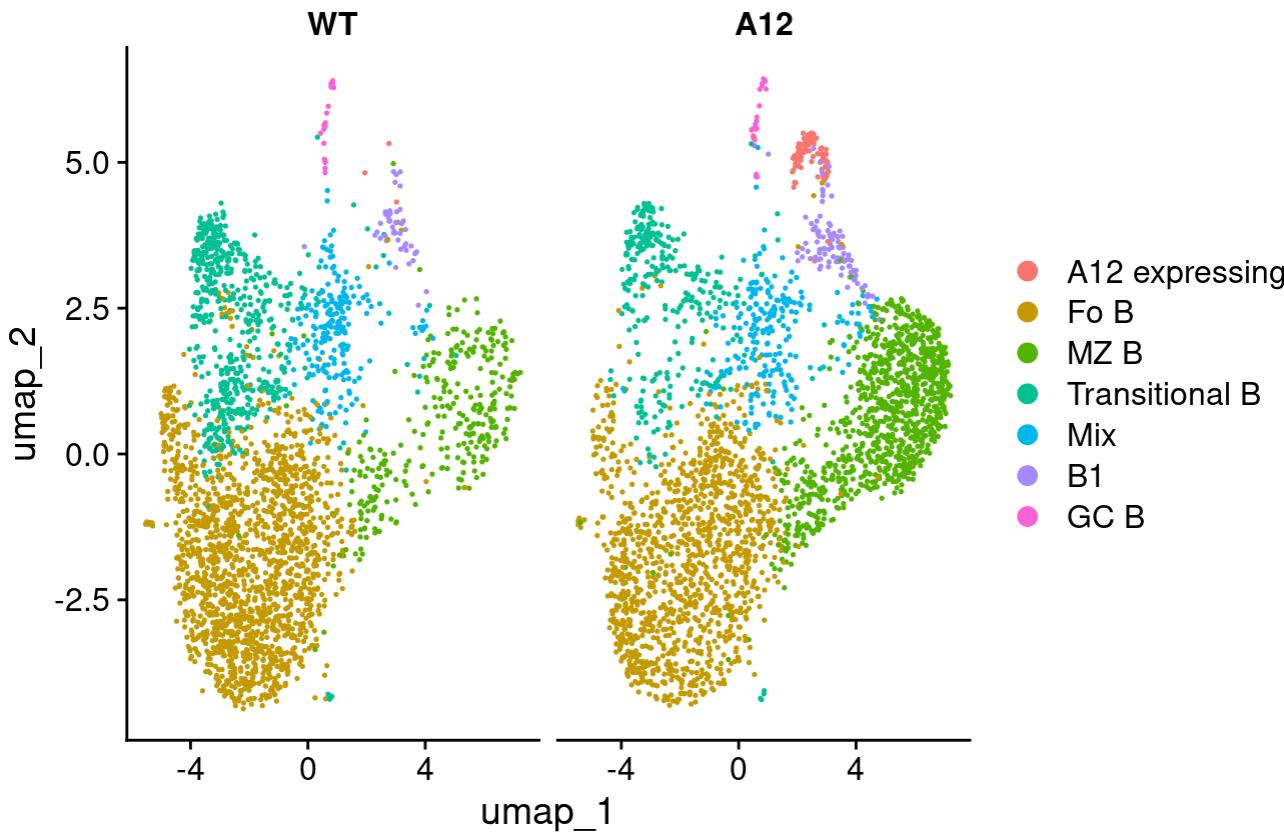
```
Idents(Seurat_Object_SP_selected_Bcells_idents, cells = A12_expressing_cells) <- "A12 expressing"
```

ng"

```
Seurat_Object_SP_selected_Bcells_idents$seurat_clusters_new_fin <- Idents(Seurat_Object_SP_selected_Bcells_idents)
```

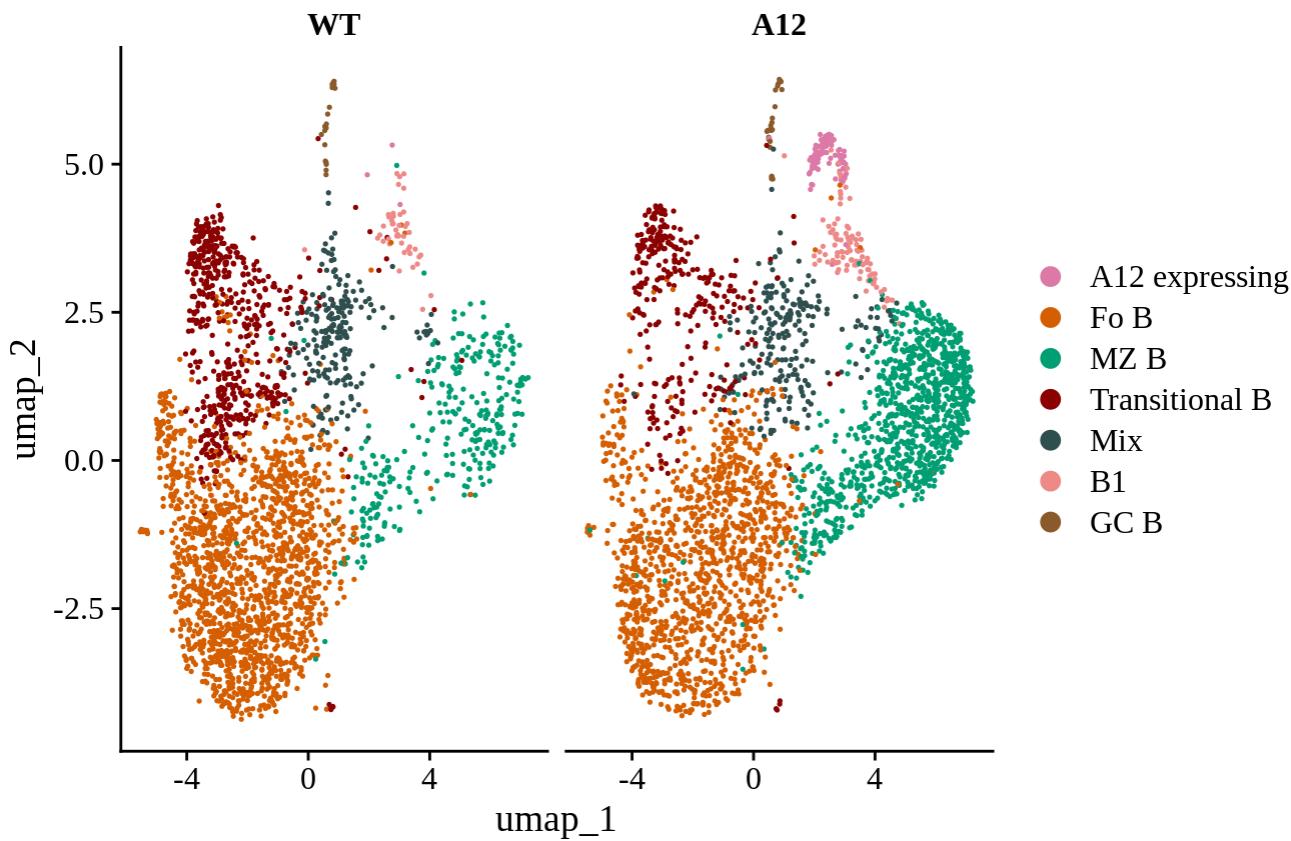
```
DimPlot(Seurat_Object_SP_selected_Bcells_idents, reduction = "umap", split.by = "genotype", group.by = "seurat_clusters_new_fin")
```

seurat_clusters_new_fin



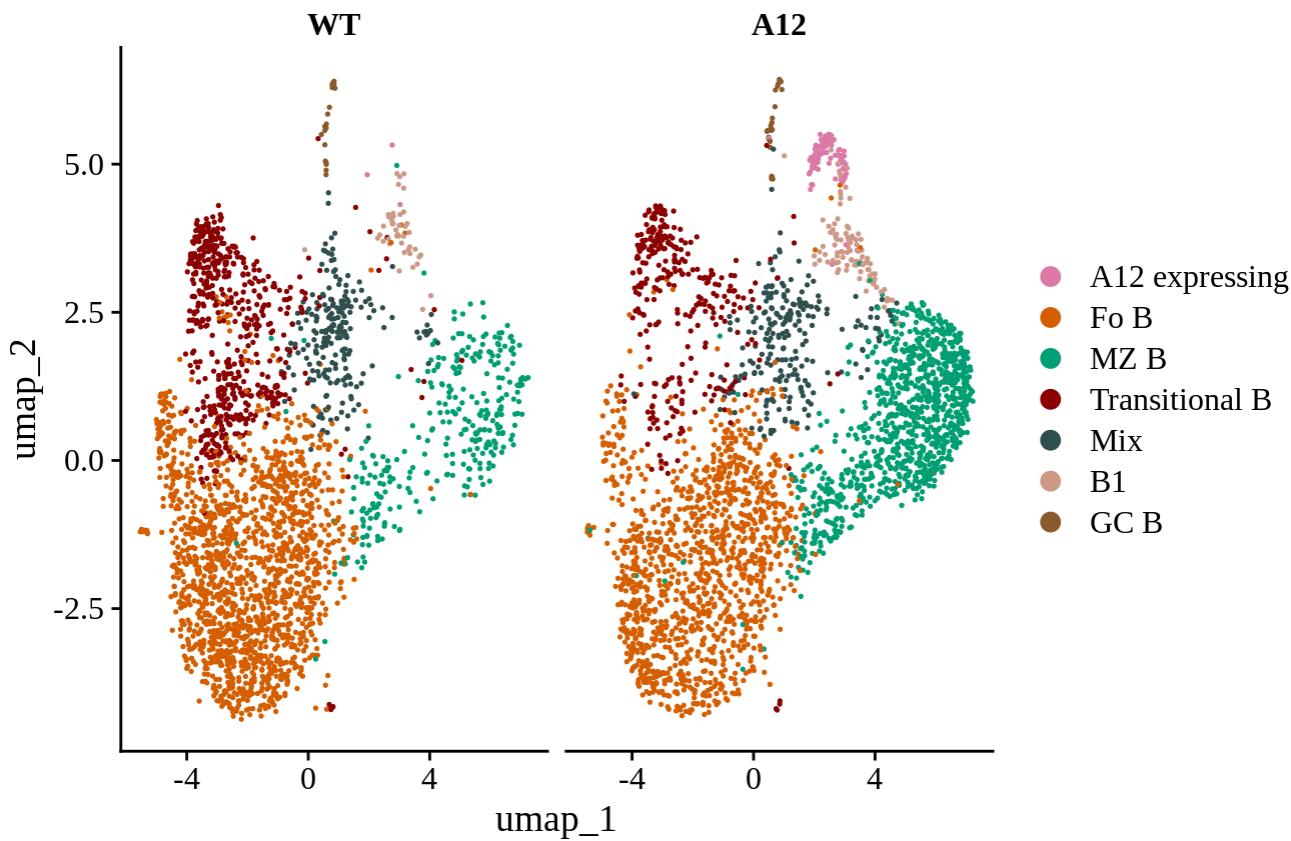
```
# Just a quick example with the built in Seurat data:
DimPlot(Seurat_Object_SP_selected_Bcells_idents, reduction = "umap", cols= c("Fo B" = "#D55E00", "Transitional B" = "darkred", "MZ B" = "#009E73", "A12 expressing" = "#DC79A7", "Mix" = "darkslategray", "GC B" = "tan4", "B1" = "#ED8987A7"), group.by = "seurat_clusters_new_fin", split.by = "genotype") +
  theme(
    text = element_text(family = "Times New Roman")
  )
```

seurat_clusters_new_fin



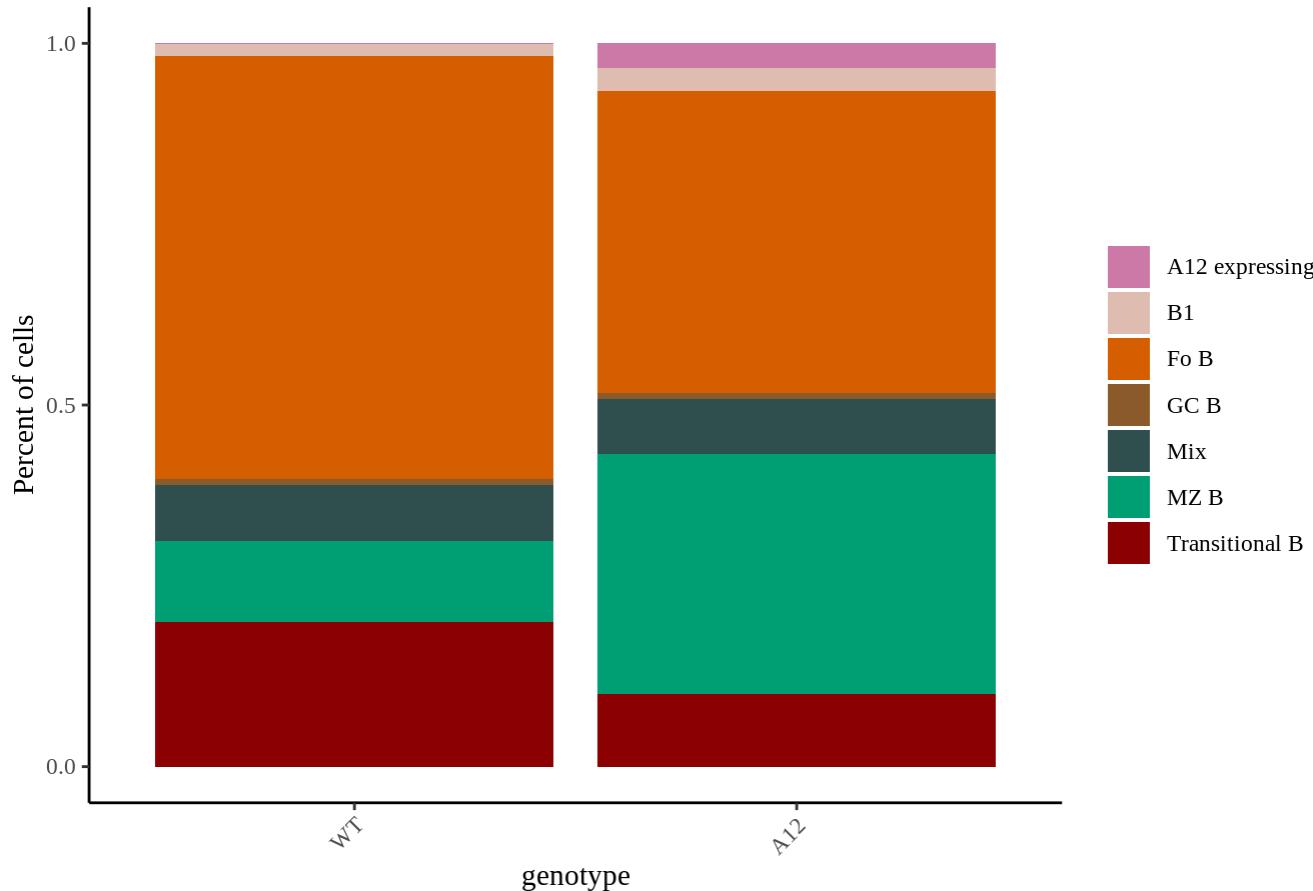
```
# Just a quick example with the built in Seurat data:
DimPlot(Seurat_Object_SP_selected_Bcells_idents, reduction = "umap", cols= c("Fo B" = "#D55E00",
"Transitional B" = "darkred", "MZ B" = "#009E73", "A12 expressing" = "#DC79A7", "Mix" = "darkslategray",
"GC B" = "tan4", "B1" = "#CD9987A7"), group.by = "seurat_clusters_new_fin", split.by = "genotype") +
  theme(
    text = element_text(family = "Times New Roman")
  )
```

seurat_clusters_new_fin



```
dittoBarPlot(
  object = Seurat_Object_SP_selected_Bcells_idents,
  scale = "percent",
  var = "seurat_clusters_new_fin",
  color.panel = c("Fo B" = "#D55E00", "Transitional B" = "darkred", "MZ B" = "#009E73", "A12
expressing" = "#CC79A7", "Mix" = "darkslategray", "GC B" = "tan4", "B1" = "#CD9987A7"),
  group.by = "genotype", x.reorder = c(2,1))+
  theme(
    text = element_text(family = "Times New Roman")
  )
```

seurat_clusters_new_fin



Statistics t-test for SP

```
# Obtain percentages for each cluster in each mouse
data_summary <- Seurat_Object_SP_selected_Bcells_idents@meta.data %>%
  group_by(genotype, mice, seurat_clusters_new_fin) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
```

`summarise()` has grouped output by 'genotype', 'mice'. You can override using
the ` `.groups` argument.

```
# Store percentages in lists for each genotype
A12_test <- list()
WT_test <- list()
for (i in unique(data_summary$seurat_clusters_new_fin)) {
  A12_test[[as.character(i)]] <- data_summary[data_summary$genotype == "A12" & data_summary$seurat_clusters_new_fin == i, ]
  WT_test[[as.character(i)]] <- data_summary[data_summary$genotype == "WT" & data_summary$seurat_clusters_new_fin == i, ]
}

# Create an empty list to store t-test results
t_test_results <- list()
```

```
# Perform t-tests for each cluster between genotypes
for (i in names(A12_test)) {
  group1 <- A12_test[[i]]$percentage
  group2 <- WT_test[[i]]$percentage

  # Perform the t-test
  result <- t.test(group1, group2)

  # Store the results in the list
  t_test_results[[paste("Comparison", i)]] <- result
}

# Rename the result names to simplify
names(t_test_results) <- sub("Comparison ", "", names(t_test_results))

# Extract p-values from the t-test results
p_values <- sapply(t_test_results, function(res) res$p.value)
p_values
```

```
## A12 expressing          Fo B          MZ B Transitional B          Mix
##      0.033333783    0.001164951    0.007658029    0.034777687    0.727346359
##          B1           GC B
##      0.034812317    0.868375851
```

```
# Calculate log fold change and add significance markers
mean_per_cluster_A12 <- sapply(A12_test, function(df) mean(df$percentage, na.rm = TRUE))
mean_per_cluster_WT <- sapply(WT_test, function(df) mean(df$percentage, na.rm = TRUE))

# Extract p-values
p_value_list <- unlist(lapply(t_test_results, function(res) res$p.value))

# Calculate Log Fold Change (LFC)
log_fold_change <- log2(mean_per_cluster_A12 / mean_per_cluster_WT)

# Create a data frame with the results
results_df <- data.frame(
  cluster = names(A12_test),
  mean_A12 = mean_per_cluster_A12,
  mean_WT = mean_per_cluster_WT,
  log_fold_change = log_fold_change,
  p_value = p_value_list
)

# Determine levels of significance or include the p-value if between 0.05 and 0.1
results_df$significance <- ifelse(
  results_df$p_value <= 0.001, "***",
  ifelse(
    results_df$p_value <= 0.01, "**",
    ifelse(
      results_df$p_value <= 0.05, "*",
      ifelse(
        results_df$p_value <= 0.1,
```

```

        sprintf("%.3f", results_df$p_value), # Format to display p-value with 3 decimals
        ""
    )
)
)
)

print(results_df)

```

| | cluster | mean_A12 | mean_WT | log_fold_change | p_value |
|-------------------|----------------|------------|------------|-----------------|-------------|
| ## A12 expressing | A12 expressing | 3.3967158 | 0.1635568 | 4.37627663 | 0.033333783 |
| ## Fo B | Fo B | 41.7801575 | 58.5145286 | -0.48597695 | 0.001164951 |
| ## MZ B | MZ B | 33.1558666 | 11.3860142 | 1.54200135 | 0.007658029 |
| ## Transitional B | Transitional B | 10.0563112 | 19.8522248 | -0.98119950 | 0.034777687 |
| ## Mix | Mix | 7.5392403 | 7.8226425 | -0.05323688 | 0.727346359 |
| ## B1 | B1 | 3.2186250 | 1.5648852 | 1.04038771 | 0.034812317 |
| ## GC B | GC B | 0.8530837 | 0.7506668 | 0.18451462 | 0.868375851 |
| | significance | | | | |
| ## A12 expressing | * | | | | |
| ## Fo B | ** | | | | |
| ## MZ B | ** | | | | |
| ## Transitional B | * | | | | |
| ## Mix | | | | | |
| ## B1 | * | | | | |
| ## GC B | | | | | |

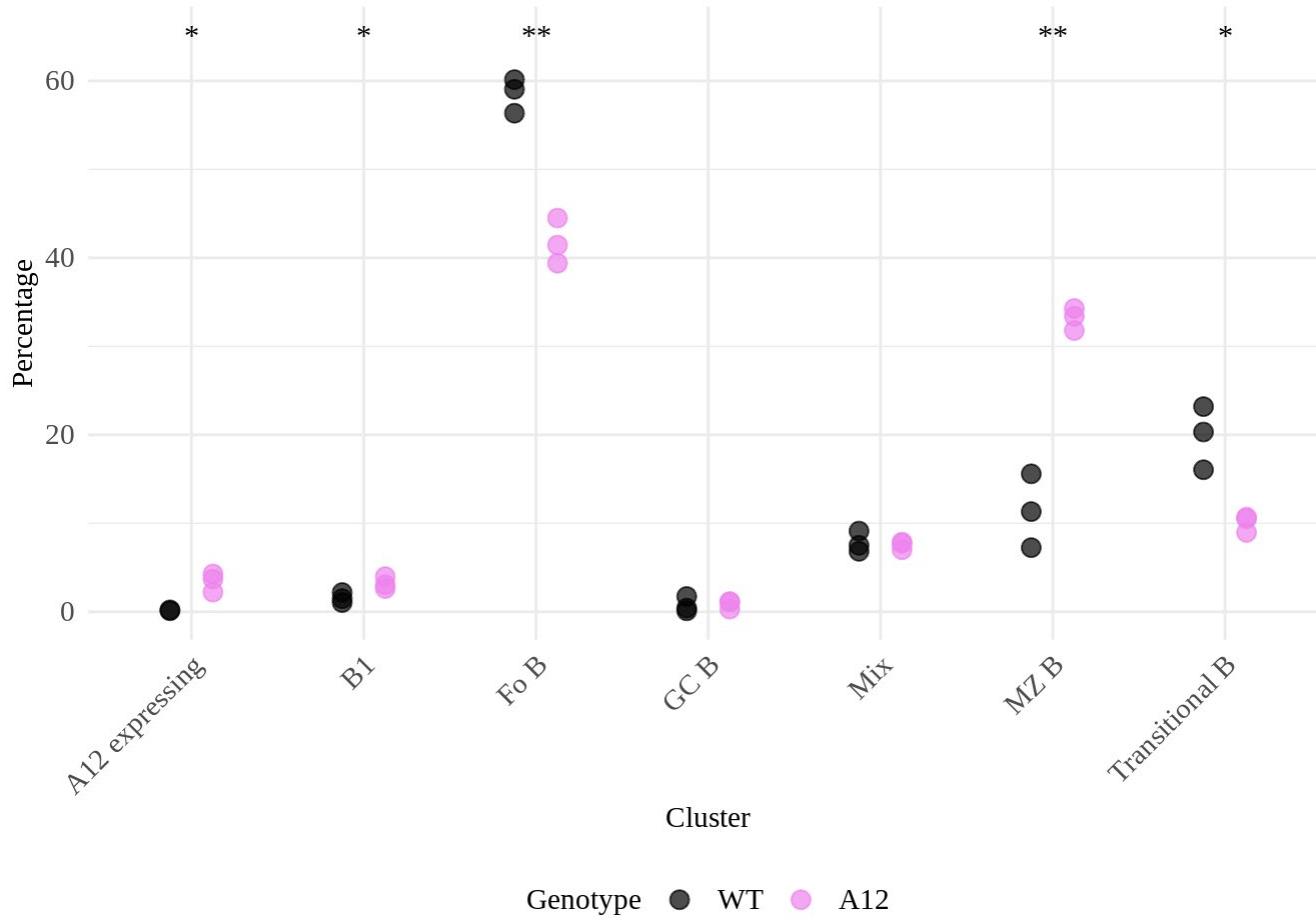
```

# Add significance levels to the summary for use in the plot
data_summary <- data_summary %>%
  left_join(results_df %>% select(cluster, significance), by = c("seurat_clusters_new_fin" = "cluster"))

# Create the plot with log fold change and significance (asterisks or p-values)
ggplot(data_summary, aes(x = seurat_clusters_new_fin, y = percentage, color = genotype)) +
  geom_point(size = 3, position = position_dodge(width = 0.5), alpha = 0.7) +
  geom_text(
    data = results_df,
    aes(x = cluster, y = max(data_summary$percentage) + 5, label = significance), # Significance or p-value
    inherit.aes = FALSE,
    size = 4,
    family = "Times New Roman"
  ) +
  scale_color_manual(values = c("WT" = "black", "A12" = "violet")) + # Custom colors
  labs(
    x = "Cluster",
    y = "Percentage",
    color = "Genotype"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(size = 16, hjust = 0.5),
    axis.text.x = element_text(size = 11, angle = 45, hjust = 1, family = "Times New Roman"),
    axis.ticks.x = element_ticks(size = 11, angle = 45, hjust = 1, family = "Times New Roman")
  )

```

```
axis.text.y = element_text(size = 11, family = "Times New Roman"),
strip.text = element_text(size = 18, family = "Times New Roman"),
legend.text = element_text(size = 11, family = "Times New Roman"),
legend.title = element_text(family = "Times New Roman"),
legend.position = "bottom"
)
```



Check difference between A12 expressing and non expressing

```
Seurat_Object_FrBC <- subset(Seurat_Object_BM_selected_Bcells_ids, seurat_clusters_new %in%
  c("ProB", "Large PreB", "Cycling ProB"))
# Create new metadata category to indicate if they express A12 or not
Seurat_Object_FrBC$A12_status <- ifelse(Seurat_Object_FrBC@assays$RNA$counts["A12", ] > 1, "A1
2_expressing", "A12_non_expressing")

table(Seurat_Object_FrBC$A12_status)
```

```
##
##      A12_expressing A12_non_expressing
##                      221                  1350
```

```
# Find relevant markers
all_markers_A12_status <- FindMarkers(
  object = Seurat_Object_FrBC,
  ident.1 = "A12_expressing",
  ident.2 = "A12_non_expressing",
  group.by = "A12_status",
  only.pos = TRUE,
  min.pct = 0.25,
  min.diff.pct = 0.1,
  thresh.use = 0.25
)

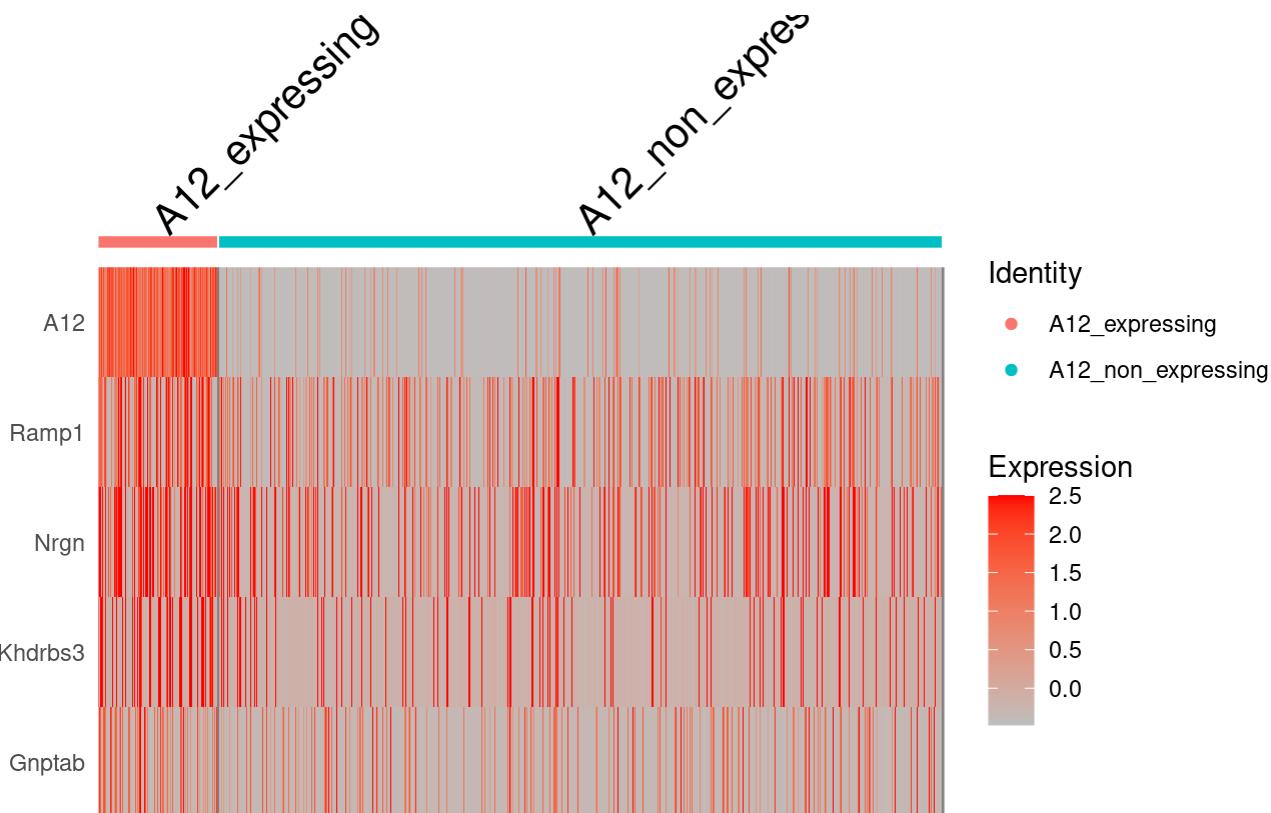
# Filter 15 most relevant genes
five_top_markers_A12_status <- all_markers_A12_status %>%
  rownames_to_column("gene") %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.3 | pct.2 > 0.3) & p_val_adj < 1e-6) %>%
  slice_min(n = 15, order_by = p_val_adj)

print(five_top_markers_A12_status)
```

| | gene | p_val | avg_log2FC | pct.1 | pct.2 | p_val_adj |
|------|---------|---------------|------------|-------|-------|---------------|
| ## 1 | A12 | 5.241424e-208 | 4.419542 | 1.000 | 0.117 | 1.042205e-203 |
| ## 2 | Ramp1 | 6.683318e-20 | 1.298866 | 0.606 | 0.320 | 1.328911e-15 |
| ## 3 | Nrgn | 1.017871e-14 | 1.169724 | 0.507 | 0.256 | 2.023935e-10 |
| ## 4 | Khdrbs3 | 2.373679e-14 | 1.219158 | 0.326 | 0.121 | 4.719824e-10 |
| ## 5 | Gnptab | 3.198818e-12 | 1.034276 | 0.398 | 0.186 | 6.360529e-08 |

```
# Heatmap
DoHeatmap(Seurat_Object_FrBC, features = five_top_markers_A12_status$gene, group.by = "A12_status") +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



```
# Average expression
average_expression_A12 <- AverageExpression(
  Seurat_Object_FrBC,
  features = five_top_markers_A12_status$gene,
  group.by = "A12_status",
  return.seurat = TRUE
)
```

```
## Names of identity class contain underscores ('_'), replacing with dashes ('-')
## This message is displayed once every 8 hours.
```

```
## Warning: The following 5 features were not found in the HTO assay: A12, Ramp1,
## Nrgn, Khdrbs3, Gnptab
```

```
## Warning: None of the features specified were found in the HTO assay.
```

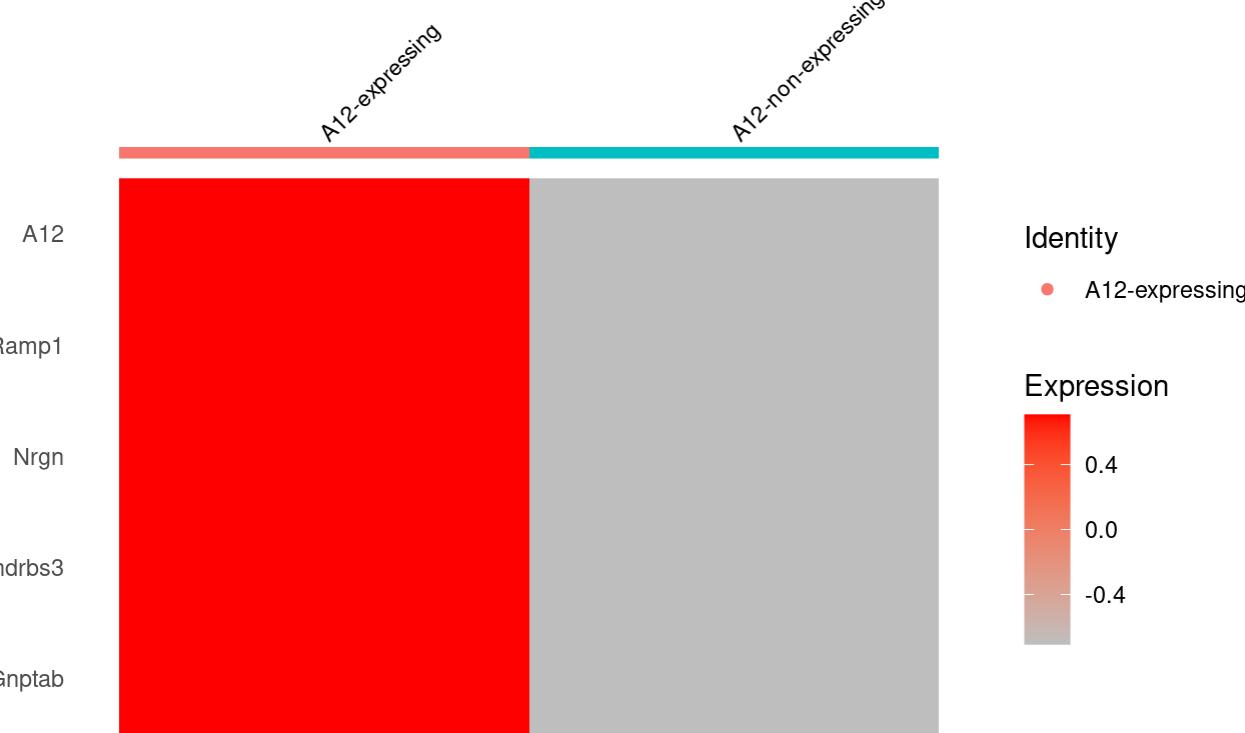
```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression_A12, features = five_top_markers_A12_status$gene, size = 3, draw.
lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red")) +
  labs(title = "Heatmap of Top Markers: A12 Expressing vs Non-Expressing")
```

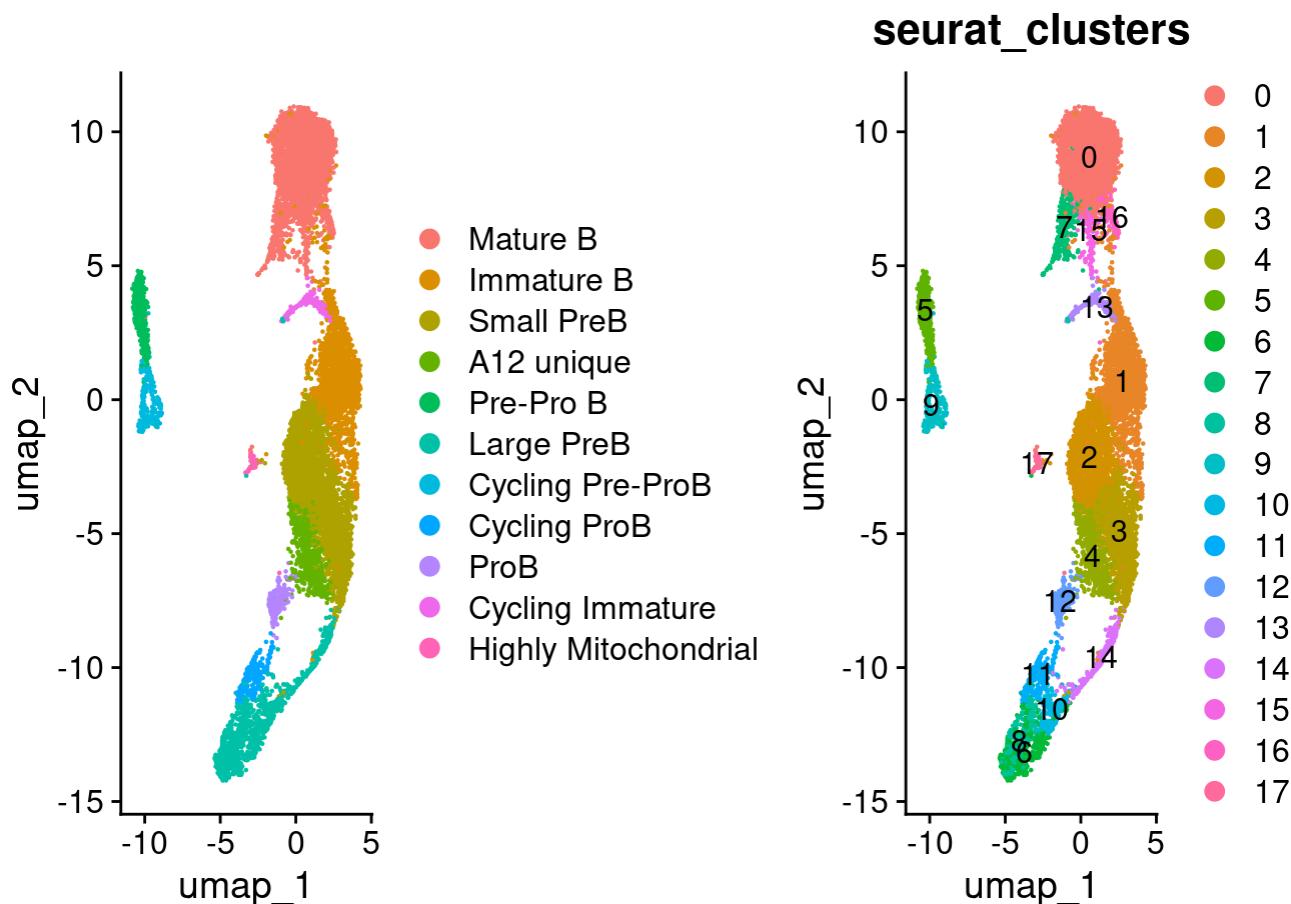
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Heatmap of Top Markers: A12 Expressing vs Non-Expressing

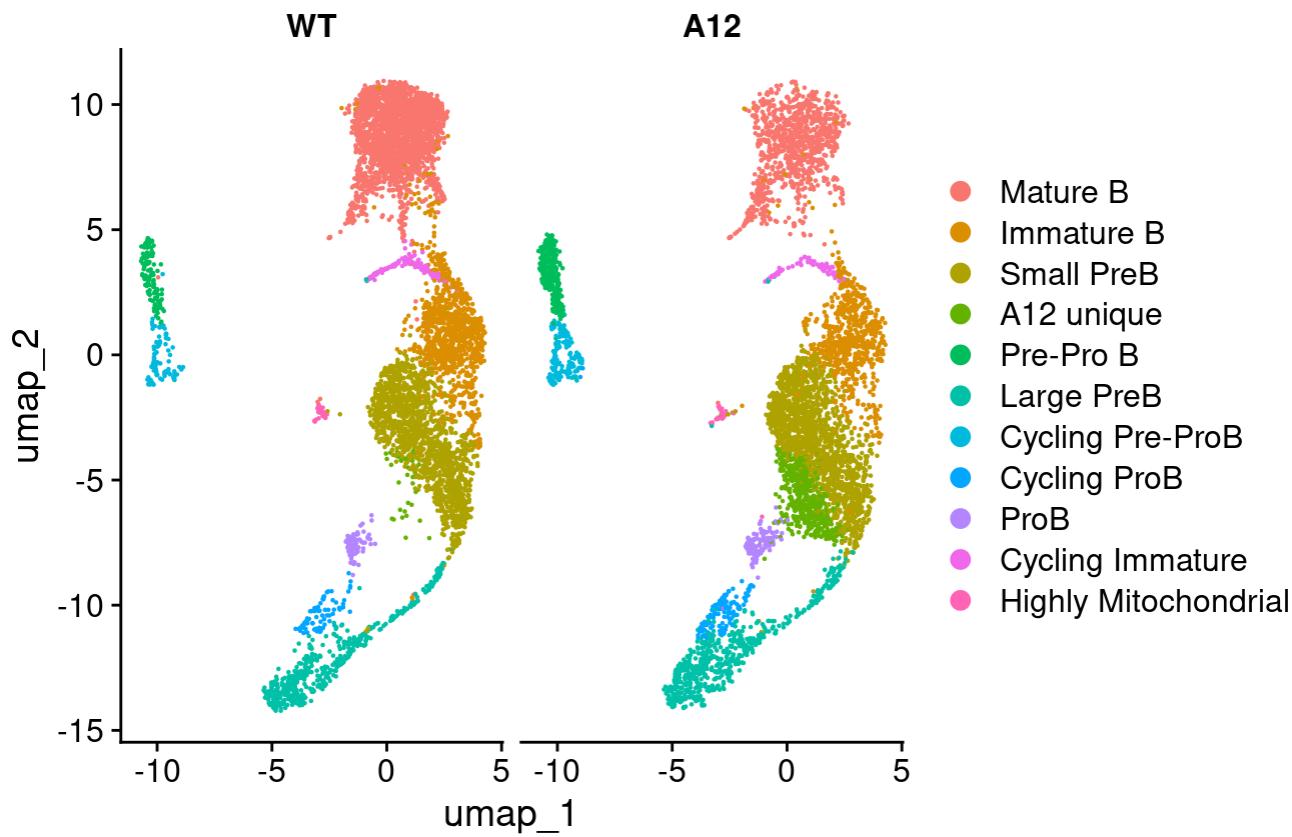


```
DimPlot(Seurat_Object_BM_selected_Bcells_idents, reduction = "umap") + DimPlot(Seurat_Object_BM_selected_Bcells_idents, reduction = "umap", group.by = "seurat_clusters", label = T)
```

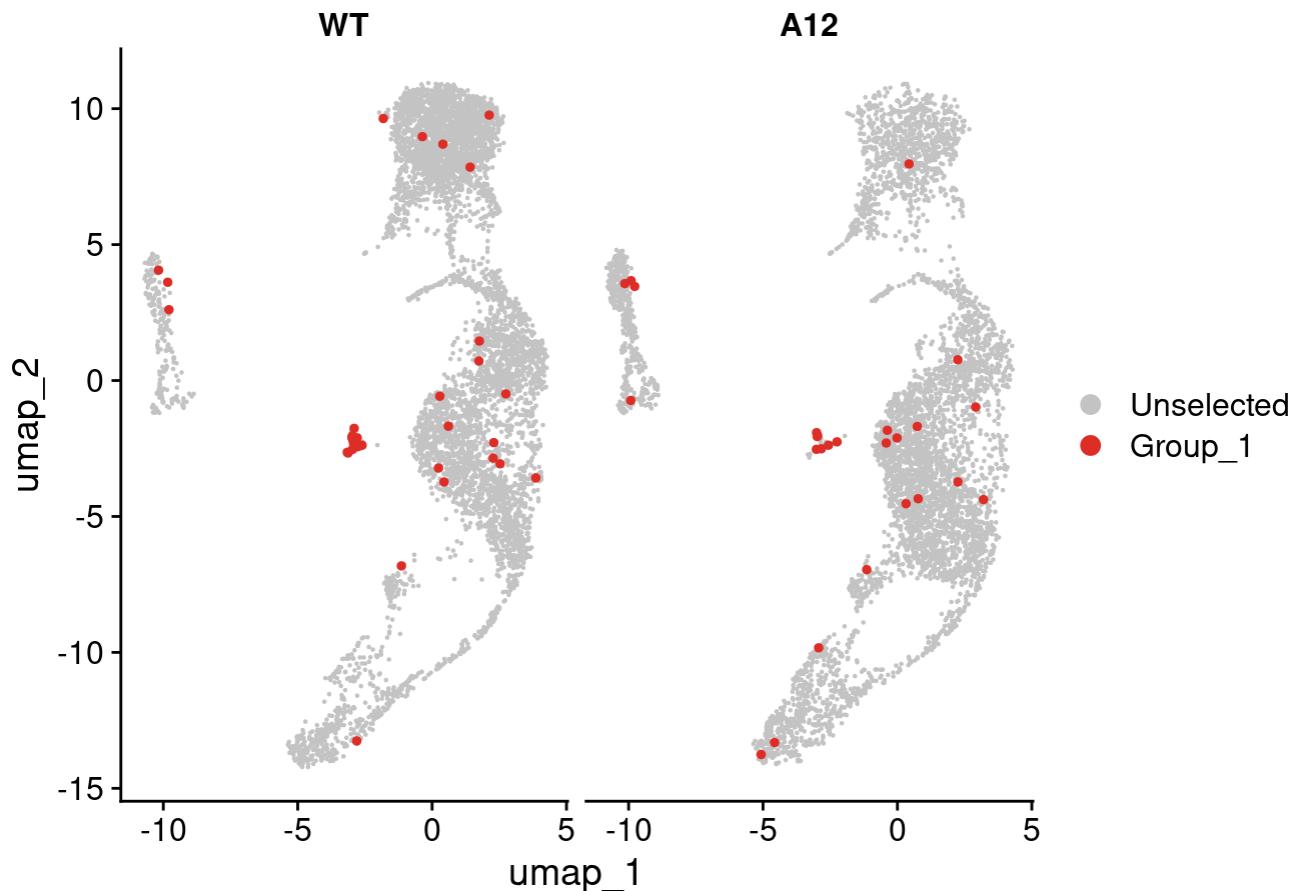


```
DimPlot(Seurat_Object_BM_selected_Bcells_idents, reduction = "umap", split.by = "genotype", group.by = "seurat_clusters_new")
```

seurat_clusters_new

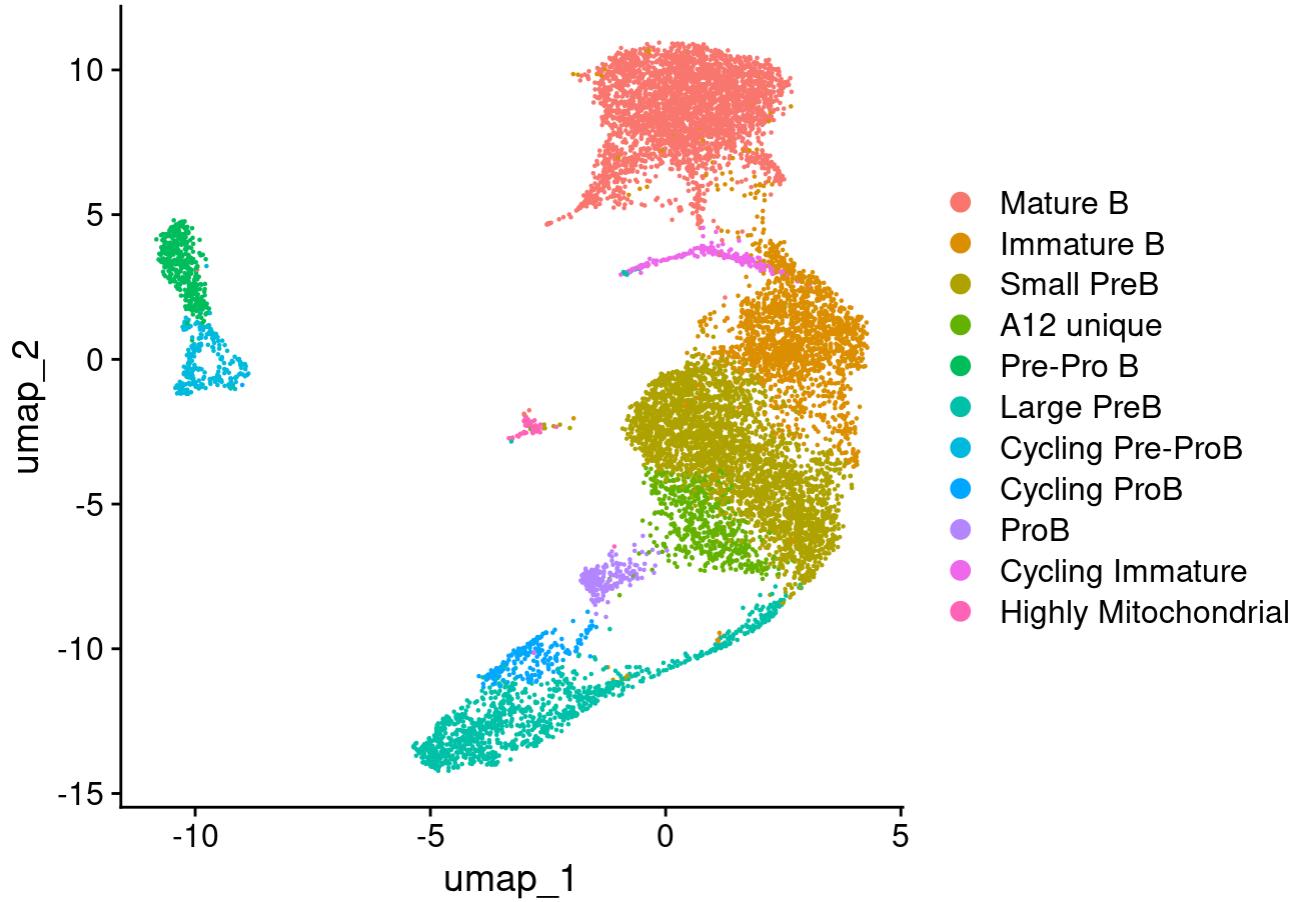


```
DimPlot(Seurat_Object_BM_selected_Bcells_idents, reduction = "umap", split.by = "genotype", cells.highlight = Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$percent.mt > 4)])
```



Bone Marrow Cluster Simplification

```
DimPlot(Seurat_Object_BM_selected_Bcells_idents)
```



```

Seurat_Object_BM_selected_Bcells_idents$simplified_clusters <- "NA"

FrA_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("Cycling Pre-ProB", "Pre-Pro B"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrA_cells, "simplified_clusters"] = "Pre-Pro B"

FrB_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("ProB", "Cycling ProB"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrB_cells, "simplified_clusters"] = "ProB"

FrD_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("Small PreB"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrD_cells, "simplified_clusters"] = "Small PreB"

A12_unique <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("A12 unique"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[A12_unique, "simplified_clusters"] = "A12 unique"

FrF_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("Mature B"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrF_cells, "simplified_clusters"] = "Mature B"

```

```

FrE_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("Immature B", "Cycling Immature"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrE_cells, "simplified_clusters"] = "Immature B"

FrC_cells <- Cells(Seurat_Object_BM_selected_Bcells_idents)[which(Seurat_Object_BM_selected_Bcells_idents$seurat_clusters_new %in% c("Large PreB"))]
Seurat_Object_BM_selected_Bcells_idents@meta.data[FrC_cells, "simplified_clusters"] = "Large PreB"

```

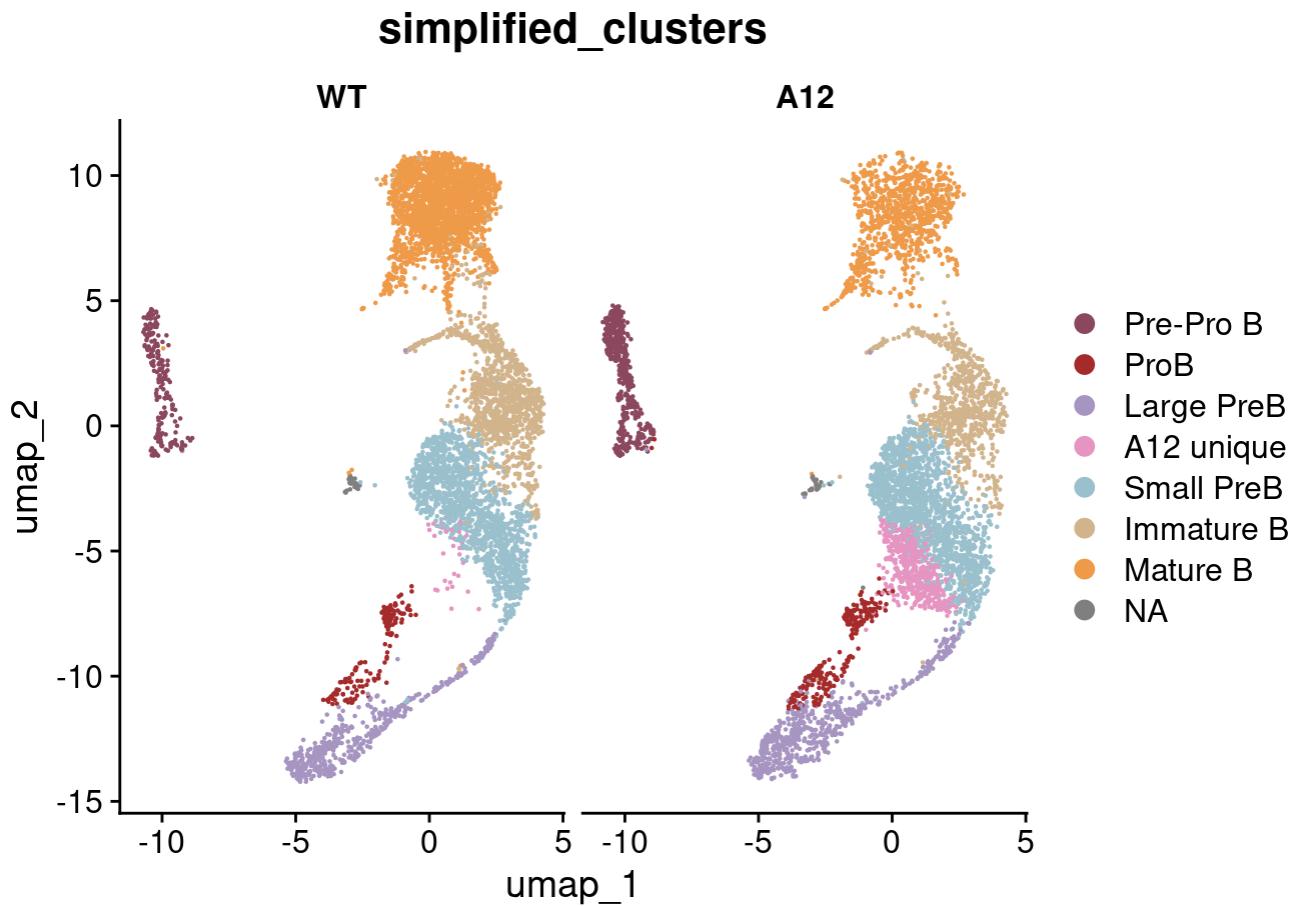
Data visualization (umaps with clusters and barplots)

```

Seurat_Object_BM_selected_Bcells_idents$simplified_clusters <-
  factor(Seurat_Object_BM_selected_Bcells_idents$simplified_clusters,
         levels = c("Pre-Pro B", "ProB", "Large PreB", "A12 unique",
                    "Small PreB", "Immature B", "Mature B"))
Seurat_Object_BM_selected_Bcells_idents$genotype <- factor(Seurat_Object_BM_selected_Bcells_idents$genotype, levels = c("WT", "A12"))

DimPlot(Seurat_Object_BM_selected_Bcells_idents, group.by = "simplified_clusters", split.by =
  "genotype", cols= c("Immature B" = "tan", "Mature B" = "tan2", "A12 unique" = "#E694C1", "Small PreB" = "lightblue3", "ProB" = "#A52A2A", "Pre-Pro B" = "palevioletred4", "Large PreB" = "#A694C1"))

```

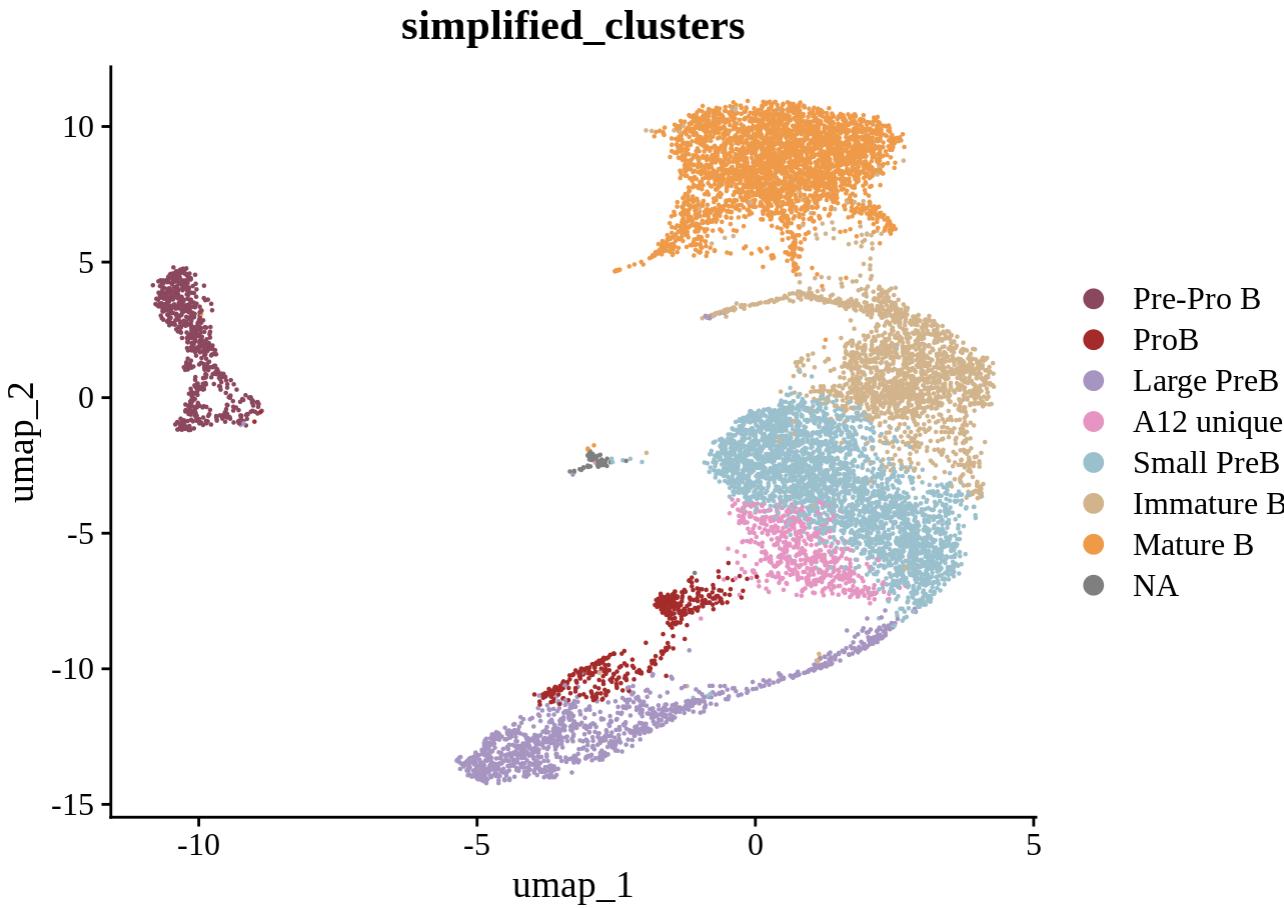


```

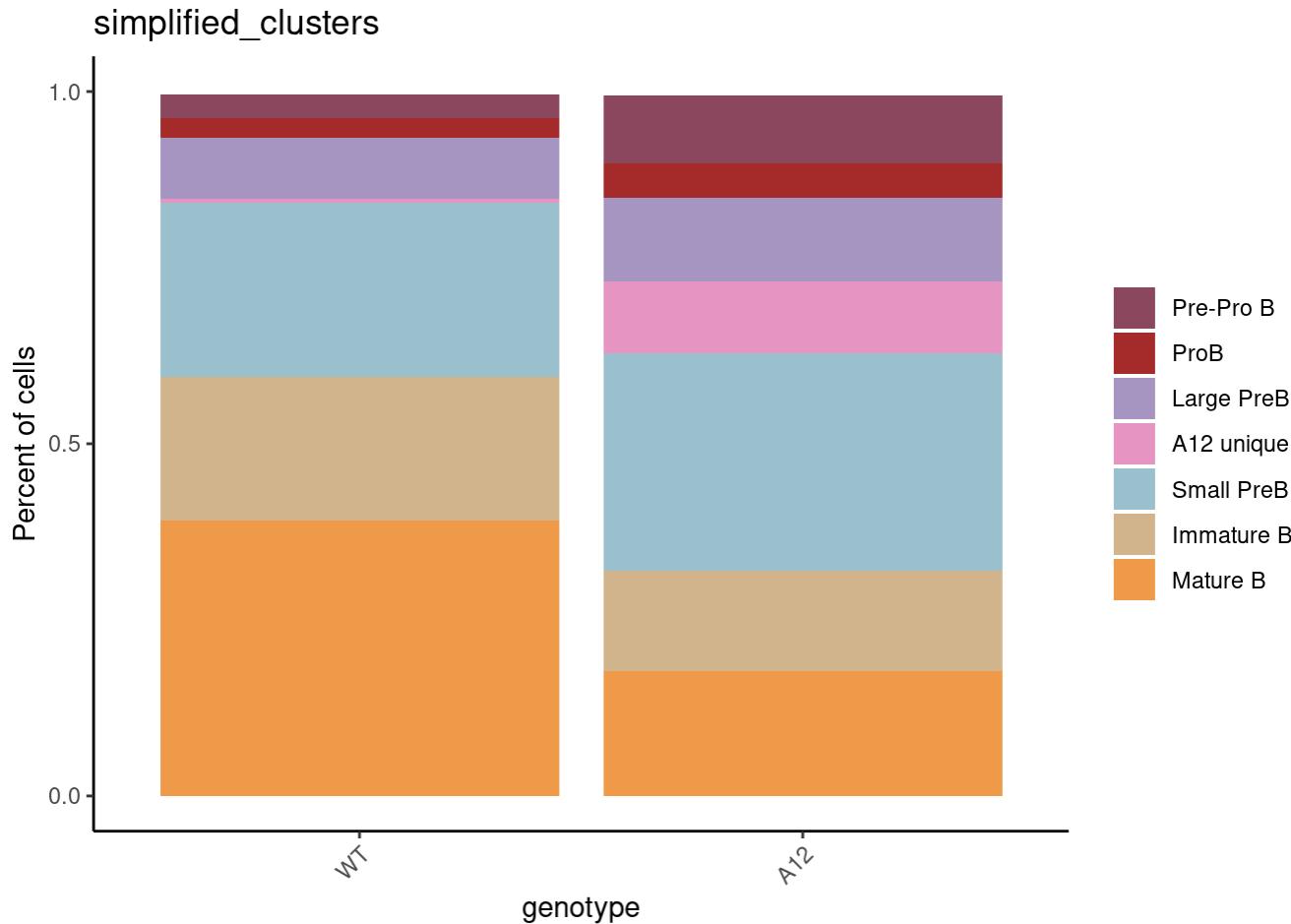
DimPlot(Seurat_Object_BM_selected_Bcells_idents, group.by = "simplified_clusters", cols= c("Im

```

```
mature_B" = "tan" , "Mature B" = "tan2", "A12 unique" = "#E694C1", "Small PreB" = "lightblue3"
, "ProB" = "#A52A2A", "Pre-Pro B" = "palevioletred4", "Large PreB" = "#A694C1")) + theme(text
= element_text(family = "Times New Roman"))
```



```
Seurat_Object_BM_selected_Bcells_idents$simplified_clusters <-
  factor(Seurat_Object_BM_selected_Bcells_idents$simplified_clusters,
         levels = c("Pre-Pro B", "ProB", "Large PreB", "A12 unique",
                   "Small PreB", "Immature B", "Mature B"))
dittoBarPlot(
  object = Seurat_Object_BM_selected_Bcells_idents,
  scale = "percent",
  var = "simplified_clusters",
  color.panel = c("Immature B" = "tan" , "Mature B" = "tan2", "A12 unique" = "#E694C1", "Small
  PreB" = "lightblue3", "ProB" = "#A52A2A", "Pre-Pro B" = "palevioletred4", "Large PreB" = "#A6
  94C1"),
  group.by = "genotype", var.labels.reorder = c(5,6,3,1,7,2,4), x.reorder = c(2,1)
)
```



Gene expression but taking into account new clusters

Wilcox test of Spleen

```

Seurat_Object_BM_A12 <- subset(Seurat_Object_BM_selected_Bcells_ids, genotype == "A12")
Seurat_Object_BM_WT <- subset(Seurat_Object_BM_selected_Bcells_ids, genotype == "WT")
Seurat_Object_SP_A12 <- subset(Seurat_Object_SP_selected_Bcells_ids, genotype == "A12")
Seurat_Object_SP_WT <- subset(Seurat_Object_SP_selected_Bcells_ids, genotype == "WT")

Idents(Seurat_Object_BM_A12) <- Seurat_Object_BM_A12$simplified_clusters
Idents(Seurat_Object_BM_WT) <- Seurat_Object_BM_WT$simplified_clusters

all_markers_SP <- FindAllMarkers(object = Seurat_Object_SP_A12, only.pos = T, min.pct = 0.25,
min.diff.pct = 0.1, thresh.use = 0.25)

```

```
## Calculating cluster A12 expressing
```

```
## Calculating cluster Fo B
```

```
## Calculating cluster MZ B
```

```
## Calculating cluster Transitional B
```

```
## Calculating cluster Mix
```

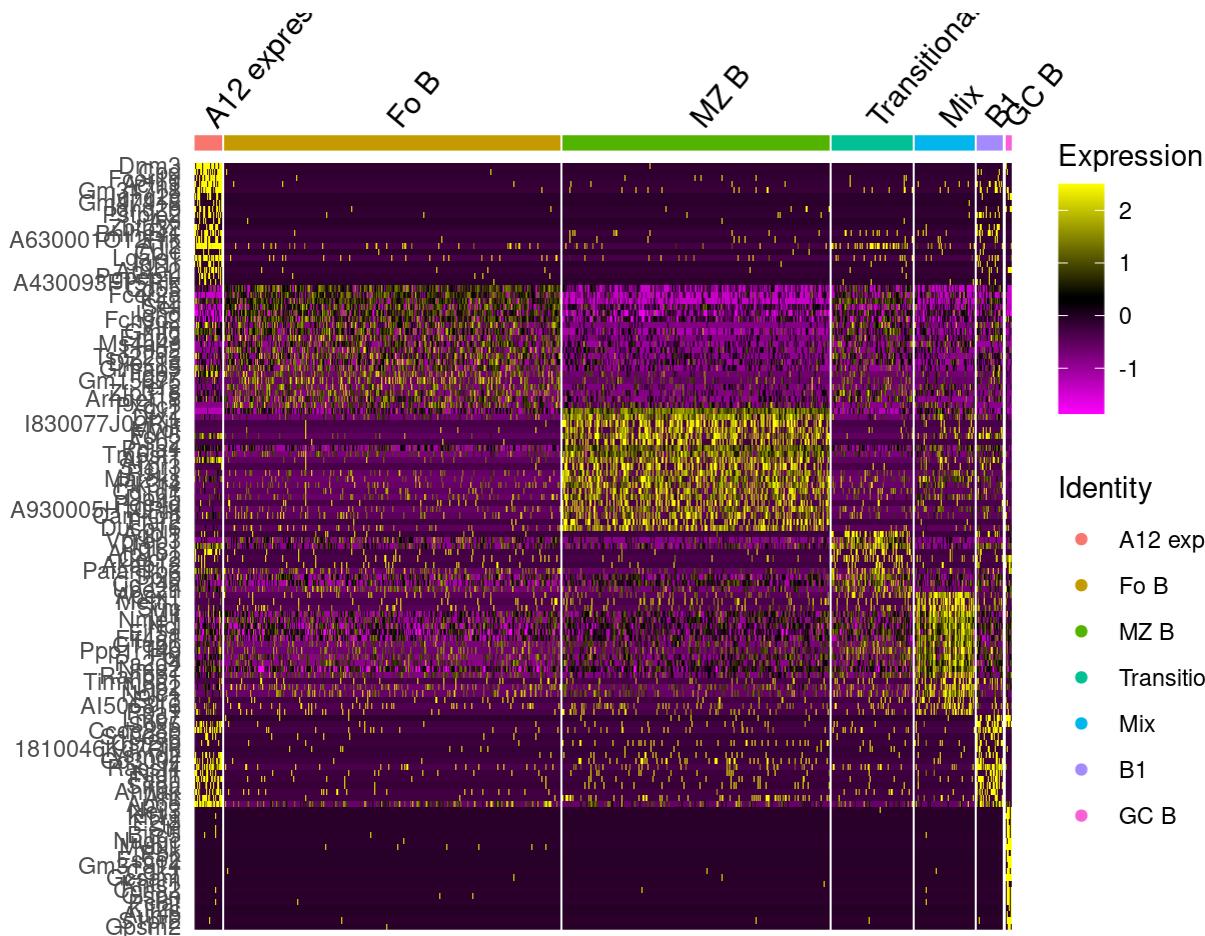
```
## Calculating cluster B1
```

```
## Calculating cluster GC B
```

We print the two most relevant per cluster:

```
five_top_markers_SP <- all_markers_SP %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.25 | pct.2 > 0.25) & p_val_adj < 1e-20) %>%
  slice_min(n = 20, order_by = p_val_adj) %>%
  ungroup() -> top5_SP
```

```
DoHeatmap(Seurat_Object_SP_A12, features = five_top_markers_SP$gene, size = 4, angle = 50)
```



```
average_expression <- AverageExpression(Seurat_Object_SP_A12, features = five_top_markers_SP$gene, return.seurat = TRUE)
```

```
## Warning: The following 125 features were not found in the HTO assay: Dnm3, Cpd,
## Fcer1g, Actn1, Gm31718, Ig hg2c, Gm47418, Ig hg2b, PstPIP2, Axl, ZbtB32, BlhE41,
## A630001O12Rik, A12, Cblc, Lgals1, Igf1r, Anxa2, Pgcep11, A430093F15Rik, Cd55,
## Fcer2a, Klf2, Sell, Ig hd, Fchsd2, Vim, Gmfg, Emp3, Ms4a4c, Lmo2, Tsc22d3,
## Dipk1a, Gimap3, Itgb7, Gm15675, Ier2, Zfp318, Arhgef18, Pxdcl, Cr2, Dtx1,
## I830077J02Rik, Myof, Fcrl5, Asb2, Pdia4, Tm6sf1, Atxn1, S1pr3, Rsu1, Marcks,
## Pik3r4, Cd1d1, Dph5, Pde4d, A930005H10Rik, Camkmt, Ffar2, Dusp16, Agbl1,
## Vpreb3, Ig lc1, Atplb1, Cecr2, Akap12, Pafahlb3, Spib, Cd24a, Ube2h, Apex1,
## Mettl1, Srm, Mif, Nme1, Ncl, Eif5a, Eif4a1, C1qbp, Ppp1r14b, Fbl, Pa2g4, Nme2,
## Ranbp1, Timm8a1, Nhp2, Nolc1, Egr3, AI506816, Ppa1, Lmo7, Sox5, Ccdc28b,
## S100a6, Csf2rb, 1810046K07Rik, Lysmd2, Cd3001f, Rassf4, Nid1, Faah, Sirpa,
## Ahnak, Ahr, Apoe, Neil3, Kif11, E2f8, Stil, Birc5, Nuggc, Myb11, Pbk, Esco2,
## Gm31814, Cdk1, Gcsam, Kntc1, Ccna2, Clspn, Pclaf, Kif2c, Aunip, S1pr2, Gpsm2
```

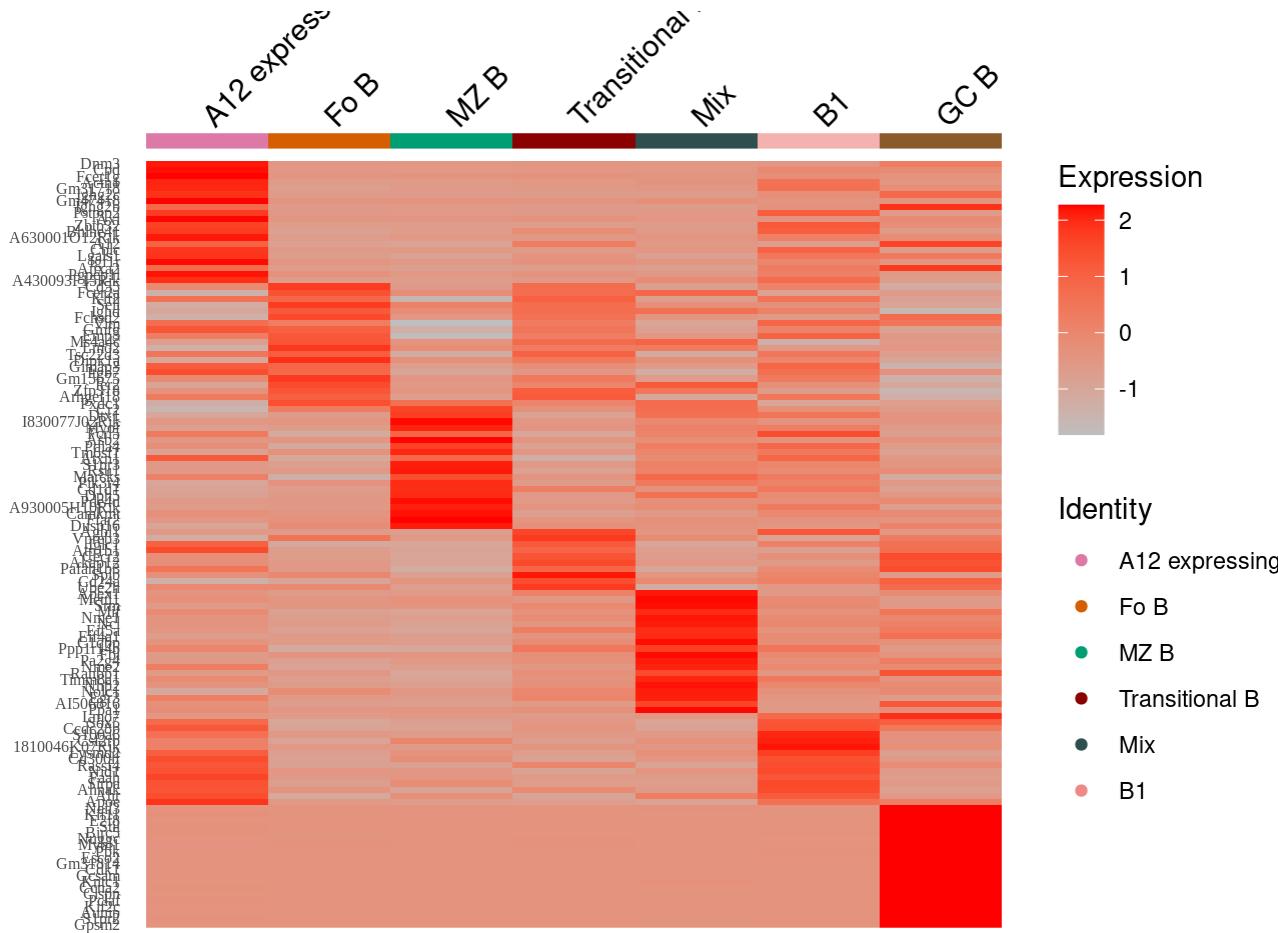
```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression, features = five_top_markers_SP$gene, group.colors = c("Fo B" = "#D55E00", "Transitional B" = "darkred", "MZ B" = "#009E73", "A12 expressing" = "#DC79A7", "Mix" = "darkslategray", "GC B" = "tan4", "B1" = "#ED8987A7"), size = 4, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red")) +
  theme(
    axis.text.y = element_text(size = 6),
    plot.title = element_text(size = 12, family = "Times New Roman"),
    plot.subtitle = element_text(size = 15, family = "Times New Roman"),
    axis.title = element_text(size = 12, family = "Times New Roman"),
    axis.text = element_text(size = 11, family = "Times New Roman")
  )
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 125 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
# Filter significant markers
five_top_markers_SP <- all_markers_SP %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.25 | pct.2 > 0.25) & p_val_adj < 1e-20) %>%
  slice_min(n = 5, order_by = p_val_adj) %>%
  ungroup() -> top5_SP

# Check top markers
print(five_top_markers_SP)
```

```
## # A tibble: 35 × 7
##   p_val avg_log2FC pct.1 pct.2 p_val_adj cluster    gene
##     <dbl>      <dbl> <dbl> <dbl>     <dbl> <fct>      <chr>
## 1 0        7.72  0.66  0.005 0   A12 expressing Dnm3
## 2 2.22e-274 6.65  0.583  0.008 3.91e-270 A12 expressing Cpd
## 3 1.32e-259 6.85  0.631  0.013 2.33e-255 A12 expressing Fcer1g
## 4 7.74e-240 5.76  0.767  0.029 1.37e-235 A12 expressing Actn1
## 5 1.07e-186 4.95  0.718  0.036 1.88e-182 A12 expressing Gm31718
## 6 7.05e-199 1.83  0.897  0.608 1.24e-194 Fo B      Cd55
## 7 1.68e-179 1.52  0.91   0.474 2.96e-175 Fo B      Fcer2a
## 8 1.41e-167 1.44  0.912  0.497 2.49e-163 Fo B      Klf2
## 9 1.72e-114 1.32  0.83   0.611 3.04e-110 Fo B      Sell
## 10 9.49e-106 1.10  0.881  0.695 1.68e-101 Fo B      Ighd
## # ... 25 more rows
```

```

# Calculate average expression for selected genes
average_expression <- AverageExpression(Seurat_Object_SP_A12, features = five_top_markers_SP$gene) $RNA %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  pivot_longer(cols = -gene, names_to = "cluster", values_to = "expression")

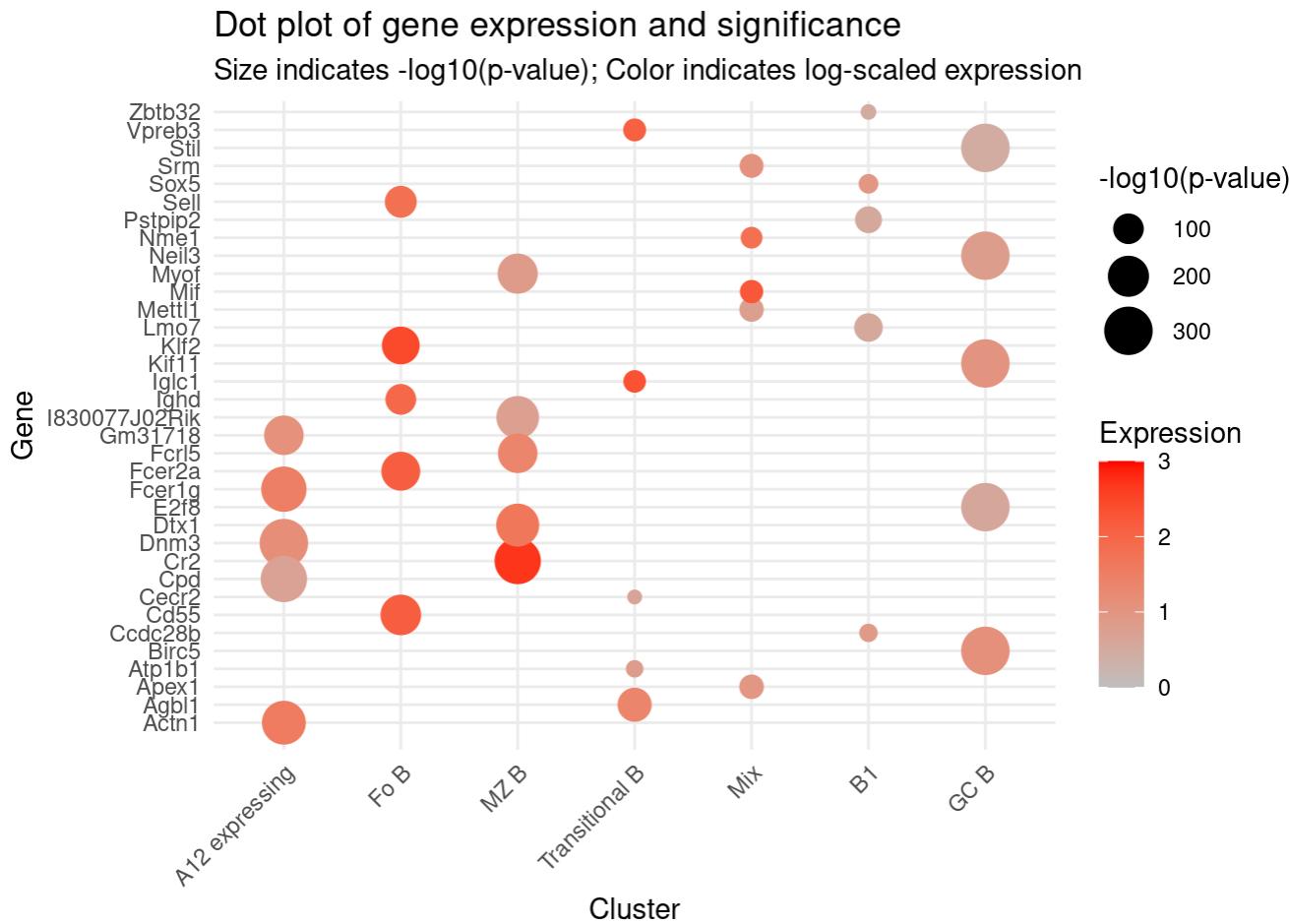
## Warning: The following 35 features were not found in the HTO assay: Dnm3, Cpd,
## Fcer1g, Actn1, Gm31718, Cd55, Fcer2a, Klf2, Sell, Ighd, Cr2, Dtx1,
## I830077J02Rik, Myof, Fcrl5, Agbl1, Vpreb3, Iglc1, Atplb1, Cecr2, Apex1, Mettl1,
## Srm, Mif, Nmeli, Lmo7, Pstpip2, Sox5, Ccdc28b, Zbtb32, Neil3, Kif11, E2f8, Stil,
## Birc5

## Warning: None of the features specified were found in the HTO assay.

# Combine expression and p-values
data_for_dotplot <- five_top_markers_SP %>%
  dplyr::select(cluster, gene, avg_log2FC, p_val_adj) %>%
  left_join(average_expression, by = c("gene", "cluster")) %>%
  mutate(
    p_val_adj = ifelse(p_val_adj == 0, 1e-300, p_val_adj),
    significance = -log10(p_val_adj),
    cluster = factor(cluster, levels = unique(cluster))
  )
# Create dotplot
# Normalize data of expression
data_for_dotplot <- data_for_dotplot %>%
  mutate(expression_scaled = log1p(expression)) # log1p(x) = log(1 + x)

ggplot(data_for_dotplot, aes(x = cluster, y = gene)) +
  geom_point(aes(size = significance, color = expression_scaled)) +
  scale_color_gradientn(
    colors = c("grey", "red"),
    name = "Expression",
    limits = c(0, 3)
  ) +
  scale_size_continuous(range = c(2, 8), name = "-log10(p-value)") +
  labs(
    x = "Cluster",
    y = "Gene",
    title = "Dot plot of gene expression and significance",
    subtitle = "Size indicates -log10(p-value); Color indicates log-scaled expression"
  ) +
  expand_limits(y = c(-0.5, length(unique(data_for_dotplot$gene)) + 0.5)) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right"
  )

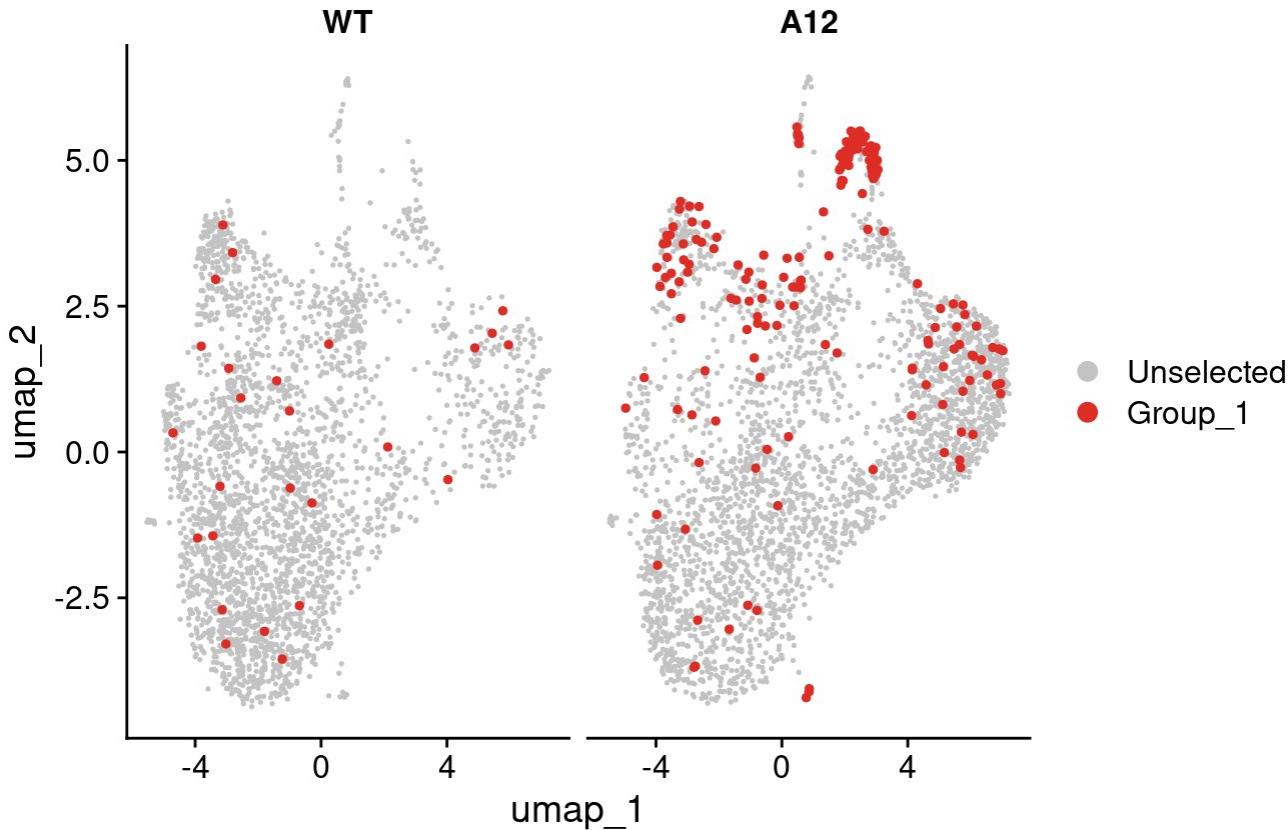
```



```
cells_to_highlight <- WhichCells(Seurat_Object_SP_selected_Bcells_idents,
                                expression = huIgk %in% c("huk_WT", "huk_A12"))

DimPlot(Seurat_Object_SP_selected_Bcells_idents,
        group.by = "seurat_clusters",
        cells.highlight = cells_to_highlight, split.by = "genotype")
```

highlight



```
positive_cells <- WhichCells(Seurat_Object_SP_A12,
                           expression = huIgk %in% c("huk_WT", "huk_A12"))
```

```
Seurat_Object_SP_A12$huIgk_positive <- ifelse(
  Cells(Seurat_Object_SP_A12) %in% positive_cells,
  "Positive",
  "Negative"
)
```

```
meta_data <- Seurat_Object_SP_A12@meta.data
```

```
percentages <- meta_data %>%
  group_by(seurat_clusters_new_fin, huIgk_positive) %>%
  summarise(n = n()) %>%
  mutate(percentage = n / sum(n) * 100)
```

`summarise()` has grouped output by 'seurat_clusters_new_fin'. You can override
using the `.groups` argument.

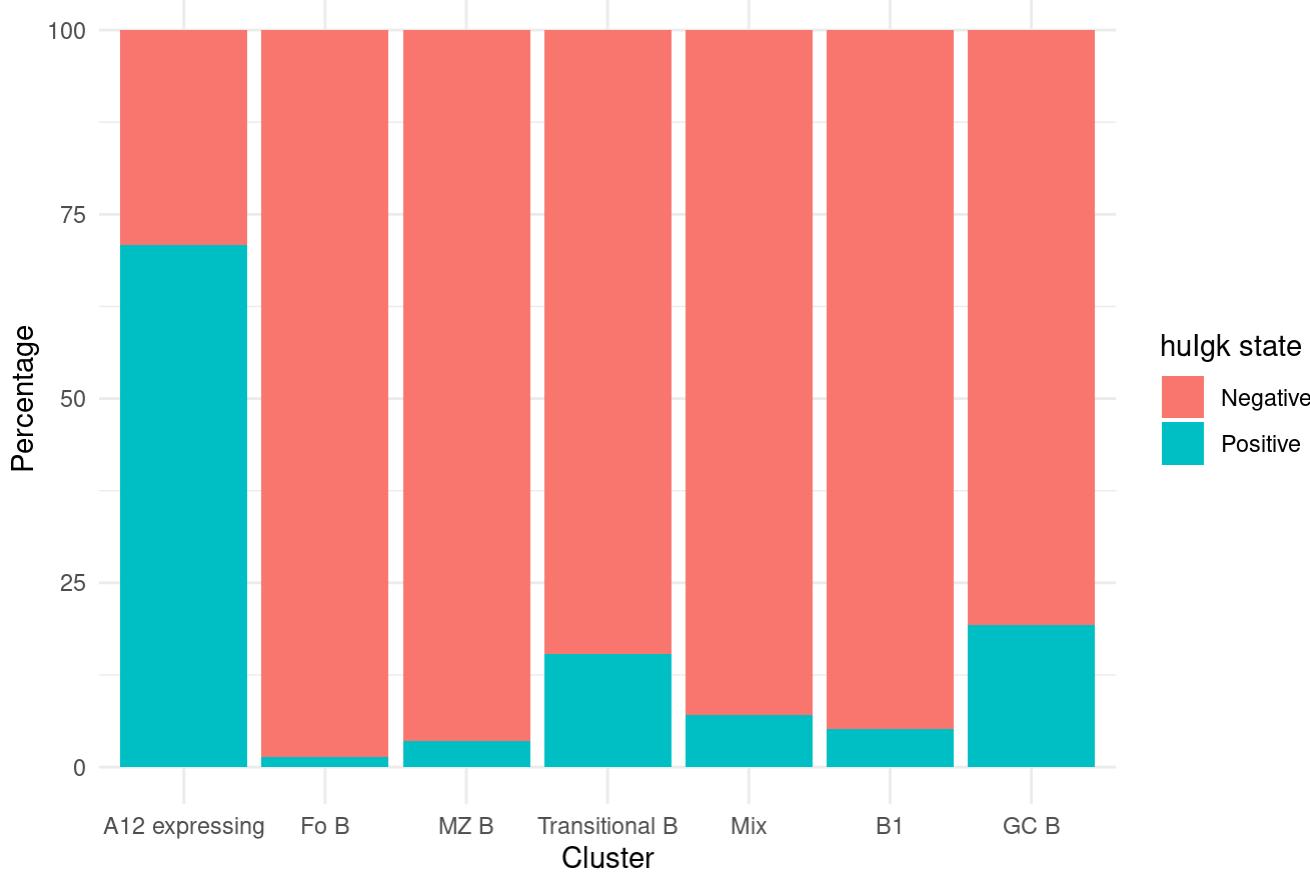
```
print(percentages)
```

```
## # A tibble: 14 × 4
## # Groups:   seurat_clusters_new_fin [7]
```

| | seurat_clusters_new_fin | huIgk_positive | n | percentage |
|-------|-------------------------|----------------|-------|------------|
| | <fct> | <chr> | <int> | <dbl> |
| ## 1 | A12 expressing | Negative | 30 | 29.1 |
| ## 2 | A12 expressing | Positive | 73 | 70.9 |
| ## 3 | Fo B | Negative | 1247 | 98.7 |
| ## 4 | Fo B | Positive | 17 | 1.34 |
| ## 5 | MZ B | Negative | 969 | 96.4 |
| ## 6 | MZ B | Positive | 36 | 3.58 |
| ## 7 | Transitional B | Negative | 258 | 84.6 |
| ## 8 | Transitional B | Positive | 47 | 15.4 |
| ## 9 | Mix | Negative | 212 | 93.0 |
| ## 10 | Mix | Positive | 16 | 7.02 |
| ## 11 | B1 | Negative | 93 | 94.9 |
| ## 12 | B1 | Positive | 5 | 5.10 |
| ## 13 | GC B | Negative | 21 | 80.8 |
| ## 14 | GC B | Positive | 5 | 19.2 |

```
ggplot(percentages, aes(x = seurat_clusters_new_fin, y = percentage, fill = huIgk_positive)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(title = "Frequency of positive cells per cluster",
       x = "Cluster",
       y = "Percentage",
       fill = "huIgk state") +
  theme_minimal()
```

Frequency of positive cells per cluster



ROC analysis in SP for A12 expressing population

```
all_markers_A12_uniq_pop <- FindMarkers(object = Seurat_Object_SP_A12, test.use = "roc", ident
  .1 = "A12 expressing", min.pct = 0.25, min.diff.pct = 0.1)
```

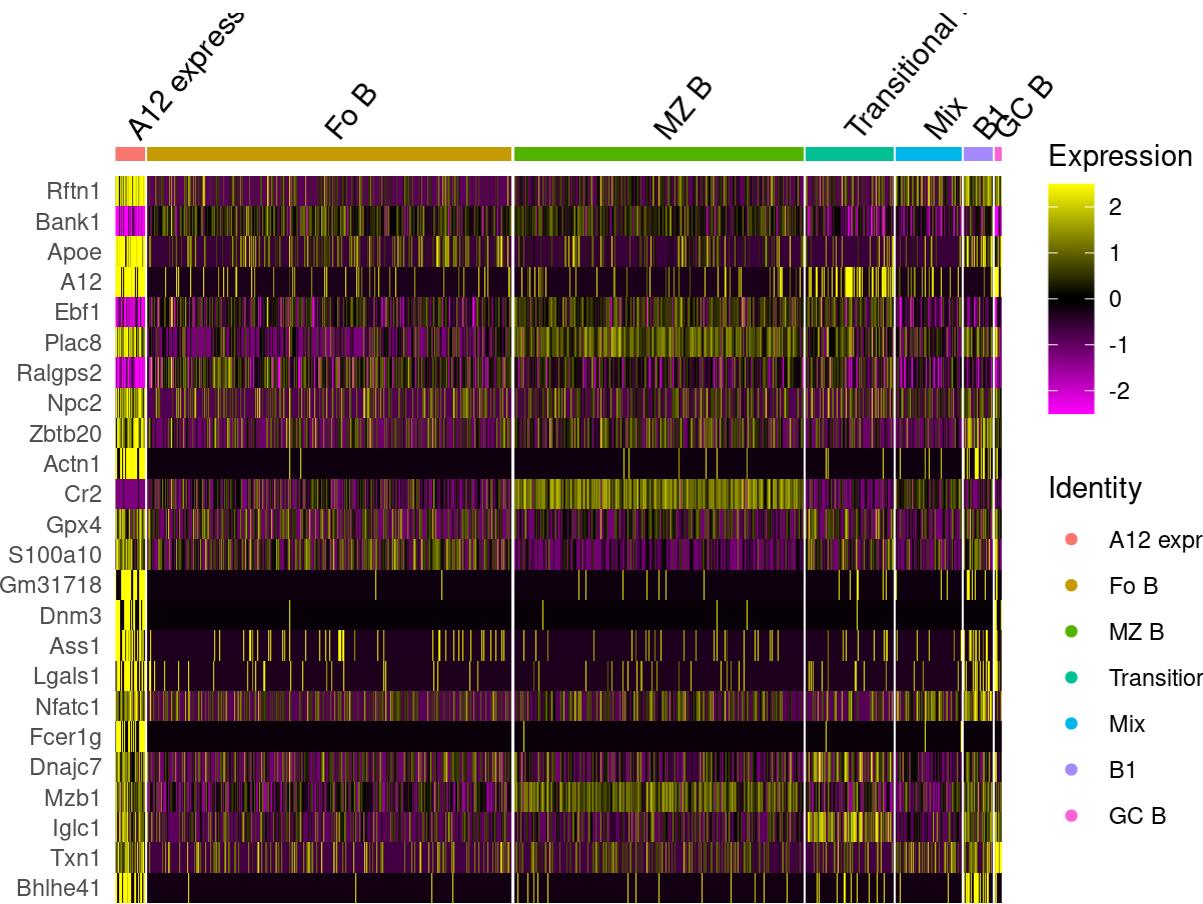
We print the two most relevant per cluster:

```
five_top_markers_A12_uniq_pop_SP <- all_markers_A12_uniq_pop %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < -1) & (power > 0.60) &
    (pct.1 > 0.25 | pct.2 > 0.25)) %>%
  slice_max(n = 30, order_by = power) %>%
  ungroup()
```

```
five_top_markers_A12_uniq_pop_SP
```

| | myAUC | avg_diff | power | avg_log2FC | pct.1 | pct.2 |
|------------|-------|------------|-------|------------|-------|-------|
| ## Rftn1 | 0.959 | 1.5166452 | 0.918 | 2.776510 | 1.000 | 0.510 |
| ## Bank1 | 0.048 | -1.5464308 | 0.904 | -2.585981 | 0.718 | 0.988 |
| ## Apoe | 0.946 | 2.6019528 | 0.892 | 4.318443 | 0.961 | 0.336 |
| ## A12 | 0.922 | 1.6492024 | 0.844 | 3.279422 | 0.922 | 0.149 |
| ## Ebf1 | 0.085 | -1.2708871 | 0.830 | -2.113749 | 0.757 | 0.990 |
| ## Plac8 | 0.901 | 1.4620940 | 0.802 | 2.306258 | 0.990 | 0.716 |
| ## Ralgps2 | 0.101 | -1.3391415 | 0.798 | -2.655489 | 0.544 | 0.929 |
| ## Npc2 | 0.888 | 1.0294094 | 0.776 | 2.182351 | 0.961 | 0.439 |
| ## Zbtb20 | 0.876 | 1.0935973 | 0.752 | 1.943452 | 0.990 | 0.620 |
| ## Actn1 | 0.874 | 1.5027901 | 0.748 | 5.760261 | 0.767 | 0.029 |
| ## Cr2 | 0.130 | -2.0562738 | 0.740 | -6.839679 | 0.039 | 0.752 |
| ## Gpx4 | 0.852 | 0.8053353 | 0.704 | 1.546531 | 0.981 | 0.612 |
| ## S100a10 | 0.847 | 1.0750714 | 0.694 | 1.908737 | 0.990 | 0.593 |
| ## Gm31718 | 0.846 | 1.0336621 | 0.692 | 4.952958 | 0.718 | 0.036 |
| ## Dnm3 | 0.828 | 1.1489419 | 0.656 | 7.715127 | 0.660 | 0.005 |
| ## Ass1 | 0.826 | 1.0360083 | 0.652 | 3.332570 | 0.738 | 0.105 |
| ## Lgals1 | 0.819 | 1.3083095 | 0.638 | 4.067962 | 0.699 | 0.086 |
| ## Nfatcl | 0.816 | 0.8393194 | 0.632 | 1.675831 | 0.922 | 0.510 |
| ## Fcer1g | 0.811 | 1.4596827 | 0.622 | 6.850361 | 0.631 | 0.013 |
| ## Dnajc7 | 0.806 | 0.8757434 | 0.612 | 1.619050 | 0.971 | 0.623 |
| ## Mzbl1 | 0.805 | 0.7224826 | 0.610 | 1.182054 | 0.990 | 0.812 |
| ## Igllc1 | 0.804 | 0.8912229 | 0.608 | 1.556116 | 0.942 | 0.538 |
| ## Txn1 | 0.802 | 0.6699049 | 0.604 | 1.615281 | 0.903 | 0.366 |
| ## Blhhe41 | 0.801 | 0.7211340 | 0.602 | 3.836938 | 0.650 | 0.047 |

```
DoHeatmap(Seurat_Object_SP_A12, features =rownames(five_top_markers_A12_uniq_pop_SP ), size =
  4, angle = 50)
```



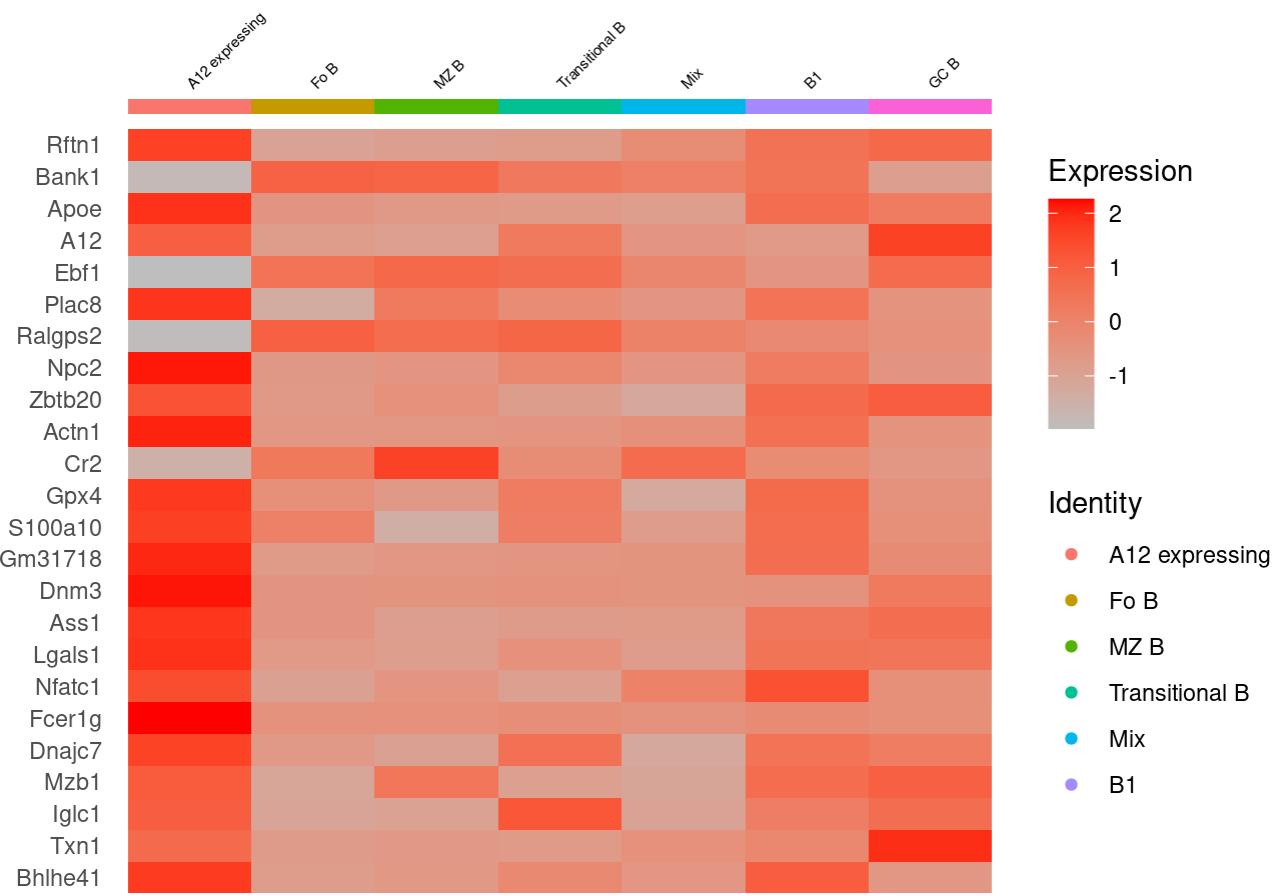
```
average_expression <- AverageExpression(Seurat_Object_SP_A12, features = five_top_markers_A12_uniq_pop_SP$gene, return.seurat = TRUE)
```

```
## Centering and scaling data matrix
```

```
DoHeatmap(average_expression, features = row.names(five_top_markers_A12_uniq_pop_SP), size = 2, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

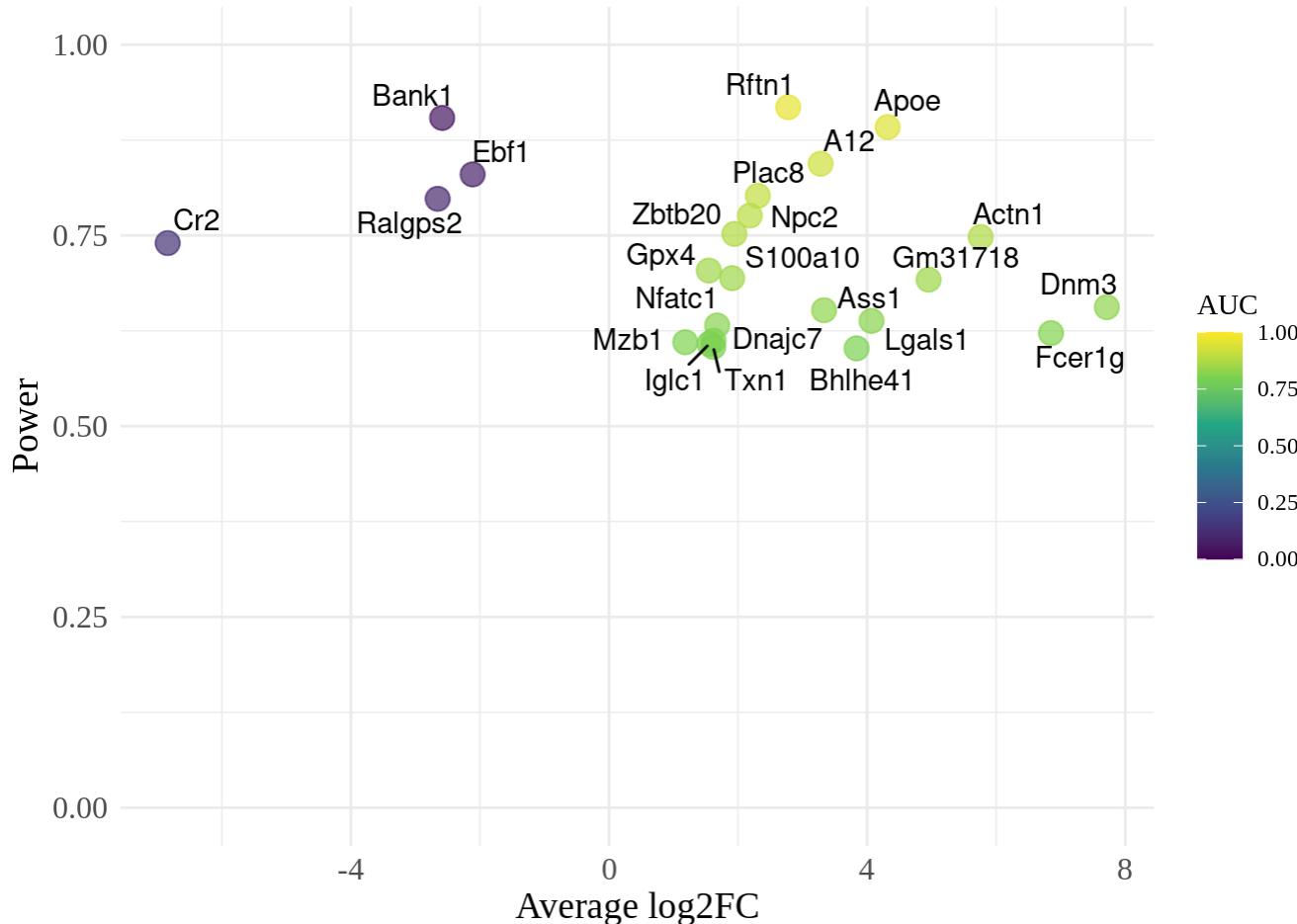
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Dotplot

```
ggplot(five_top_markers_A12_uniq_pop_SP, aes(x = avg_log2FC, y = power, label = row.names(five_top_markers_A12_uniq_pop_SP))) +
  geom_point(aes(color = myAUC), alpha = 0.7, size = 4) +
  geom_text_repel() # Etiquetas de genes
  scale_y_continuous(limits = c(0, 1), name = "Power") +
  scale_color_viridis_c(option = "D", name = "AUC", limits = c(0, 1)) +
  labs(
    x = "Average log2FC"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14)
  )
}
```



Mast analysis for A12 expressing in spleen

```
all_markers_A12_uniq_pop <- FindMarkers(object = Seurat_Object_SP_A12, test.use = "MAST", iden.t.1 = "A12 expressing", min.pct = 0.25, min.diff.pct = 0.1)
```

```
##  
## Done!
```

```
## Combining coefficients and standard errors
```

```
## Calculating log-fold changes
```

```
## Calculating likelihood ratio tests
```

```
## Refitting on reduced model...
```

```
##  
## Done!
```

DotPlot and Heatmap

```

five_top_markers_A12_uniq_pop_SP <- all_markers_A12_uniq_pop %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < 1) & p_val_adj < 1e-30 &
    (pct.1 > 0.25 | pct.2 > 0.25)) %>%
  slice_min(n = 30, order_by = p_val_adj) %>%
  ungroup() %>%
  rownames_to_column("gene")

average_expression <- AverageExpression(Seurat_Object_SP_A12, features = five_top_markers_A12_
  uniq_pop_SP$gene)$RNA %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  pivot_longer(cols = -gene, names_to = "cluster", values_to = "expression")

```

```

## Warning: The following 30 features were not found in the HTO assay: Apoe,
## Rftn1, Bank1, Dnm3, Actn1, Ebf1, Fcer1g, Cpd, Plac8, A12, Gm31718, Npc2,
## Zbtb20, Ralgps2, Lgals1, Bhlhe41, Cr2, Ass1, Pstpip2, Ig hg2b, S100a10, Aldh3b1,
## Gpx4, Ig hg2c, Zbtb32, A630001O12Rik, Gm47418, Kcnn4, Dnajc7, Anxa2

```

```

## Warning: None of the features specified were found in the HTO assay.

```

```

DoHeatmap(Seurat_Object_SP_A12, features = average_expression$gene, size = 2, draw.lines = FALSE)
  + scale_fill_gradientn(colors = c("grey", "red"))

```

```

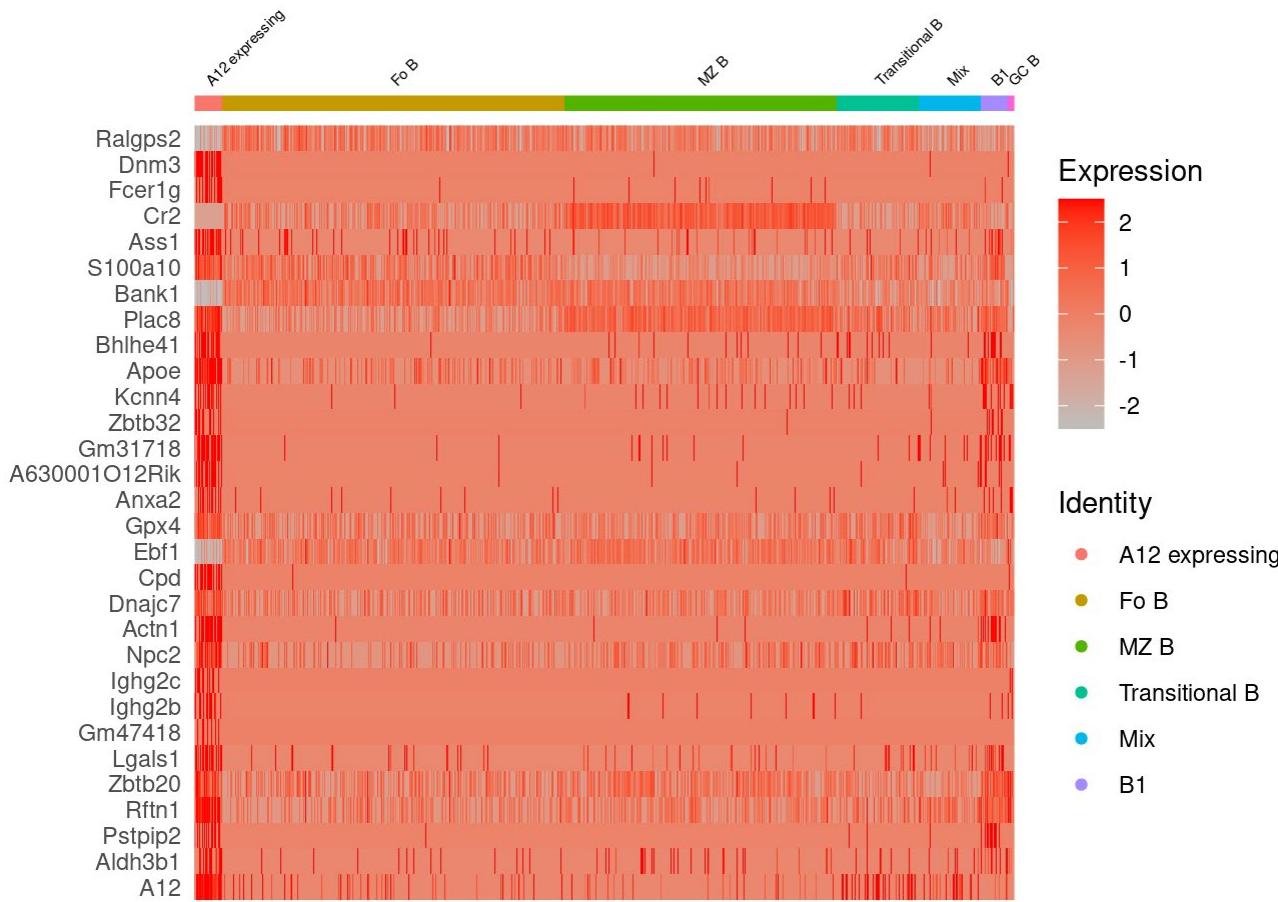
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

```

```

## Warning: Removed 780 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



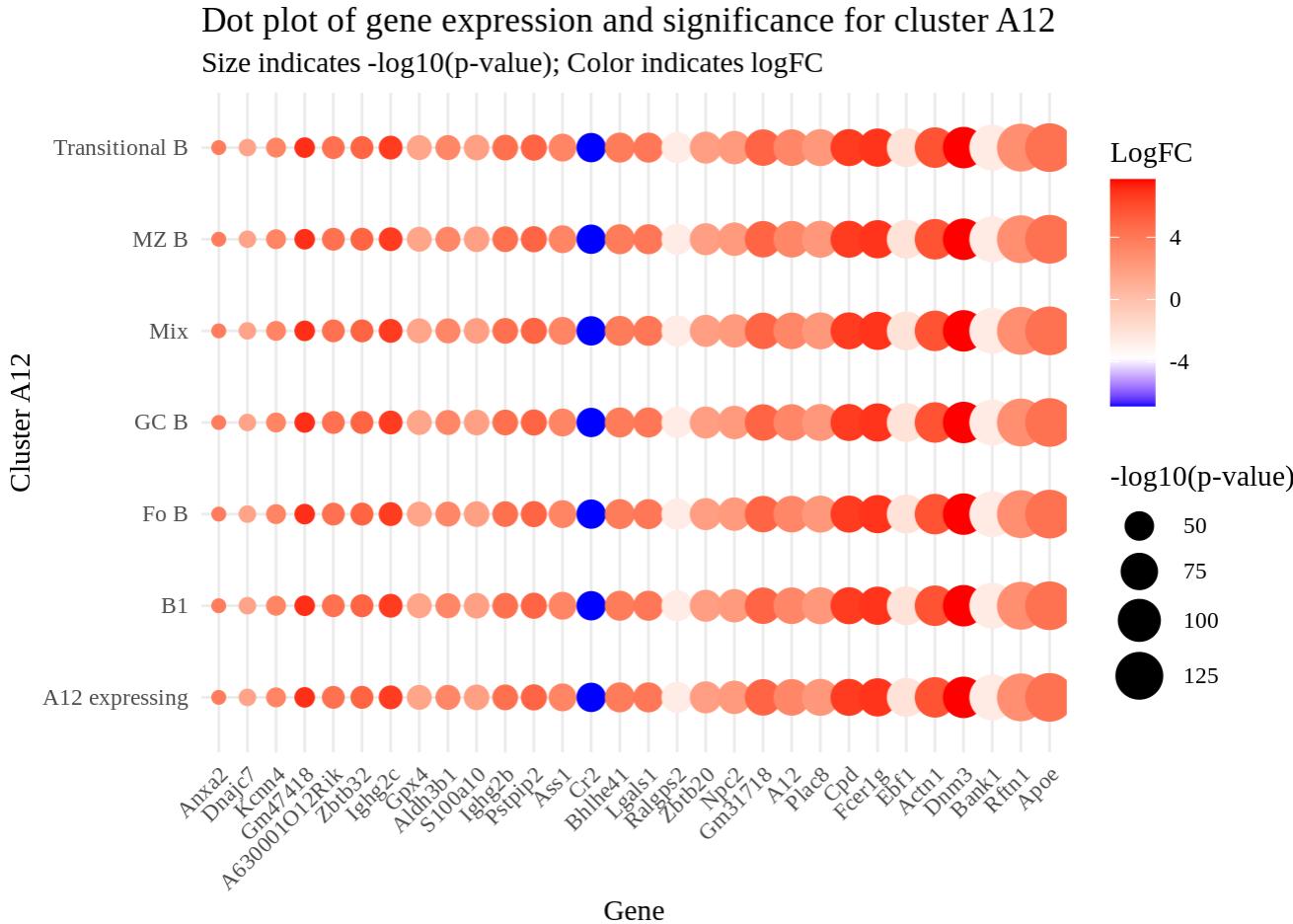
```

data_for_dotplot <- five_top_markers_A12_uniq_pop_SP %>%
  dplyr::select(gene, avg_log2FC, p_val_adj) %>%
  left_join(average_expression, by = "gene") %>%
  mutate(
    p_val_adj = ifelse(p_val_adj == 0, 1e-300, p_val_adj),
    significance = -log10(p_val_adj)
  )

data_for_dotplot_A12 <- data_for_dotplot %>%
  arrange(significance) %>%
  mutate(gene = factor(gene, levels = unique(gene)))

ggplot(data_for_dotplot_A12, aes(x = gene, y = cluster)) +
  geom_point(aes(size = significance, color = avg_log2FC)) +
  scale_color_gradientn(
    colors = c("blue", "white", "red"),
    values = scales::rescale(c(-2, 0, 7.5)),
    name = "LogFC"
  ) +
  scale_size_continuous(range = c(2, 8), name = "-log10(p-value)") +
  labs(
    x = "Gene",
    y = "Cluster A12",
    title = "Dot plot of gene expression and significance for cluster A12",
    subtitle = "Size indicates -log10(p-value); Color indicates logFC"
  )
  
```

```
theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, family = "Times New Roman"),
    text = element_text(family = "Times New Roman"),
    legend.position = "right"
  )
```



Bone Marrow Wilcox Test

```
Idents(Seurat_Object_BM_A12) <- Seurat_Object_BM_A12$simplified_clusters
all_markers_BM <- FindAllMarkers(object = Seurat_Object_BM_A12, only.pos = T, min.pct = 0.25,
min.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Calculating cluster Pre-Pro B
```

```
## Calculating cluster ProB
```

```
## Calculating cluster Large PreB
```

```
## Calculating cluster A12 unique
```

```
## Calculating cluster Small PreB
```

```
## Calculating cluster Immature B
```

```
## Calculating cluster Mature B
```

We print the two most relevant per cluster:

```
five_top_markers_BM <- all_markers_BM %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1 & (pct.1 > 0.25 | pct.2 > 0.25) & p_val_adj < 1e-20) %>%
  slice_min(n = 20, order_by = p_val_adj, with_ties = FALSE) %>%
  ungroup()

five_top_markers_BM
```

```
## # A tibble: 140 × 7
##   p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##   <dbl>    <dbl> <dbl> <dbl>    <dbl> <fct> <chr>
## 1 0     8.78 0.988 0.014    0 Pre-Pro B Siglech
## 2 0     7.91 0.984 0.02     0 Pre-Pro B Runx2
## 3 0     6.40 0.984 0.034    0 Pre-Pro B Fcer1g
## 4 0     7.65 0.961 0.015    0 Pre-Pro B Mpeg1
## 5 0     7.78 0.955 0.011    0 Pre-Pro B Fyb
## 6 0     6.91 0.94  0.03     0 Pre-Pro B Ctsl
## 7 0     8.55 0.908 0.007    0 Pre-Pro B Cd7
## 8 0     6.41 0.934 0.034    0 Pre-Pro B Alox5ap
## 9 0     6.32 0.982 0.086    0 Pre-Pro B Cox6a2
## 10 0    5.97 0.885 0.034    0 Pre-Pro B Tex2
## # ... i 130 more rows
```

```
average_expression <- AverageExpression(Seurat_Object_BM_A12, assays = "RNA", features = five_top_markers_BM$gene, return.seurat = TRUE)
```

```
## Warning: Removing 27 cells missing data for vars requested
```

```
## Removing cells with NA for 1 or more grouping variables
## Centering and scaling data matrix
```

```
DoHeatmap(
  average_expression,
  features = five_top_markers_BM$gene,
  size = 3,
  draw.lines = FALSE,
  group.colors = c(
    "Immature B" = "tan",
    "Mature B" = "tan2",
    "A12 unique" = "#E694C1",
    "Small PreB" = "lightblue3",
```

```

"ProB" = "#A52A2A",
"Pre-Pro B" = "palevioletred4",
"Large PreB" = "#A694C1"
)
) +
scale_fill_gradientn(colors = c("grey", "red")) +
theme(
  axis.text.y = element_text(size = 5),
  plot.title = element_text(size = 12, family = "Times New Roman"),
  plot.subtitle = element_text(size = 8, family = "Times New Roman"),
  axis.title = element_text(size = 12, family = "Times New Roman"),
  axis.text = element_text(size = 11, family = "Times New Roman")
)

```

```

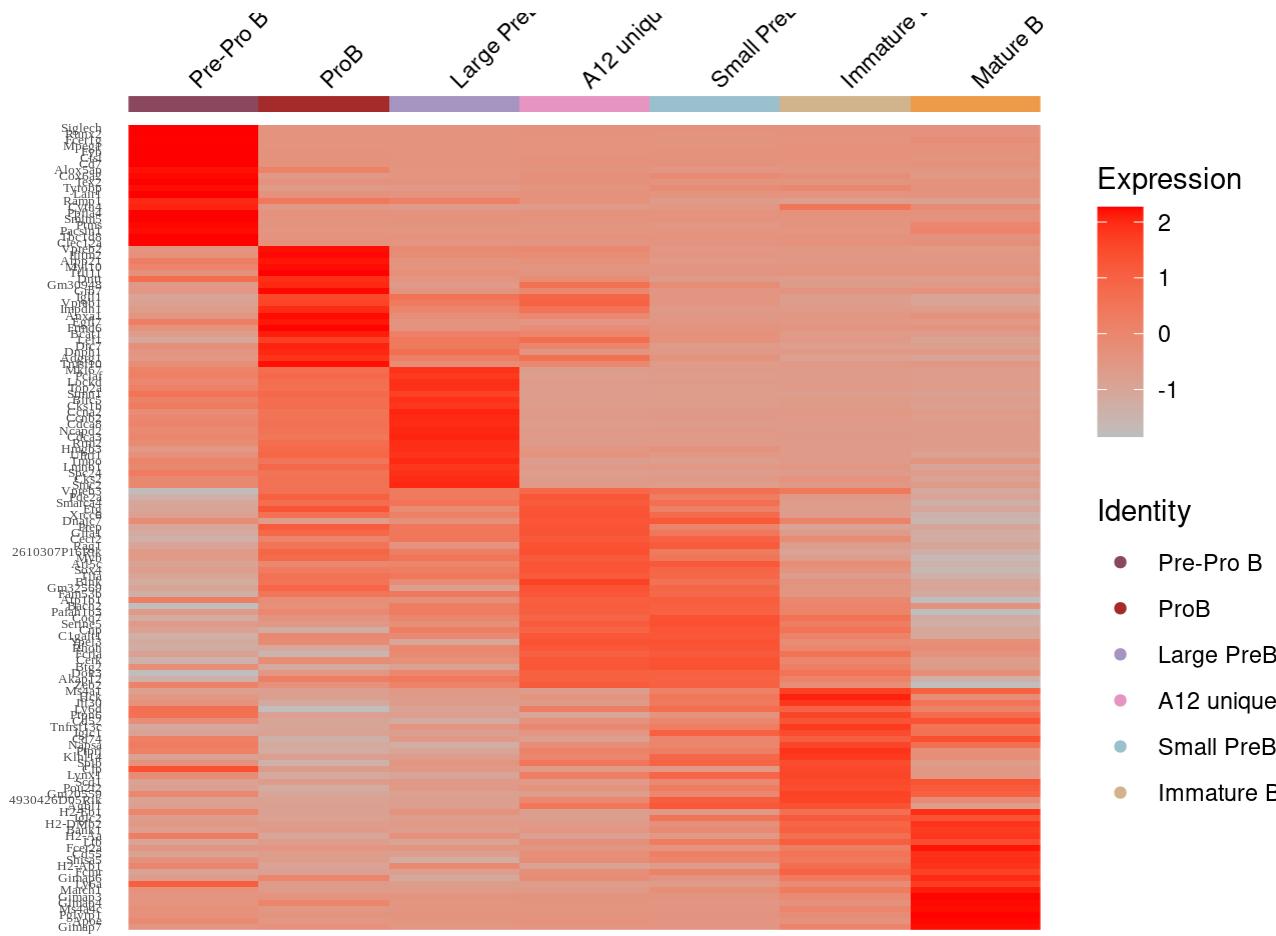
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

```

```

## Warning: Removed 133 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



CITE-seq in Bone Marrow

```

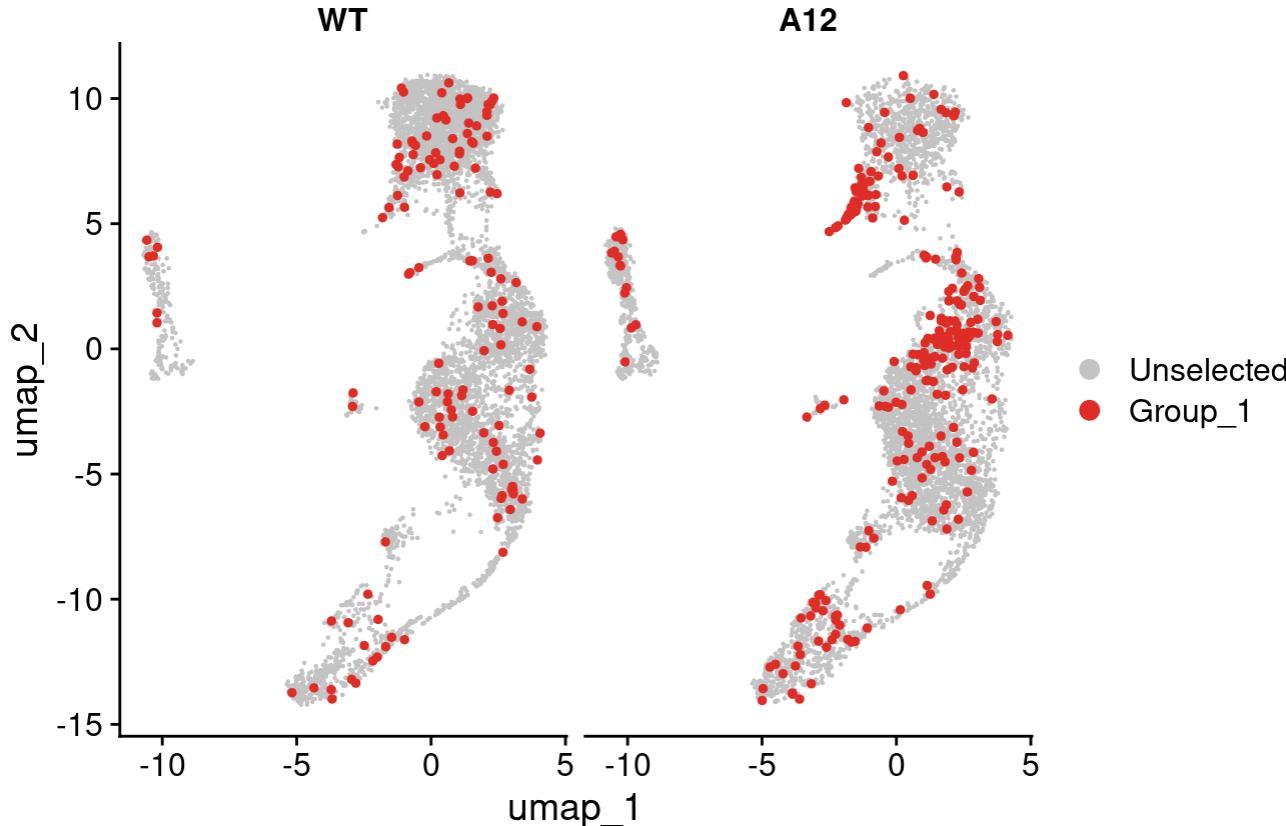
cells_to_highlight <- WhichCells(Seurat_Object_BM_selected_Bcells_idents,
                                expression = huIgk %>% c("huk WT", "huk A12"))

```

in

```
DimPlot(Seurat_Object_BM_selected_Bcells_idents,
        group.by = "seurat_clusters",
        cells.highlight = cells_to_highlight, split.by = "genotype")
```

highlight



```
positive_cells <- WhichCells(Seurat_Object_BM_A12,
                             expression = huIgk %in% c("huk_WT", "huk_A12"))
```

```
Seurat_Object_BM_A12$huIgk_positive <- ifelse(
  Cells(Seurat_Object_BM_A12) %in% positive_cells,
  "Positive",
  "Negative"
)
```

```
meta_data <- Seurat_Object_BM_A12@meta.data
```

```
percentages <- meta_data %>%
  group_by(seurat_clusters_new, huIgk_positive) %>%
  summarise(n = n()) %>%
  mutate(percentage = n / sum(n) * 100)
```

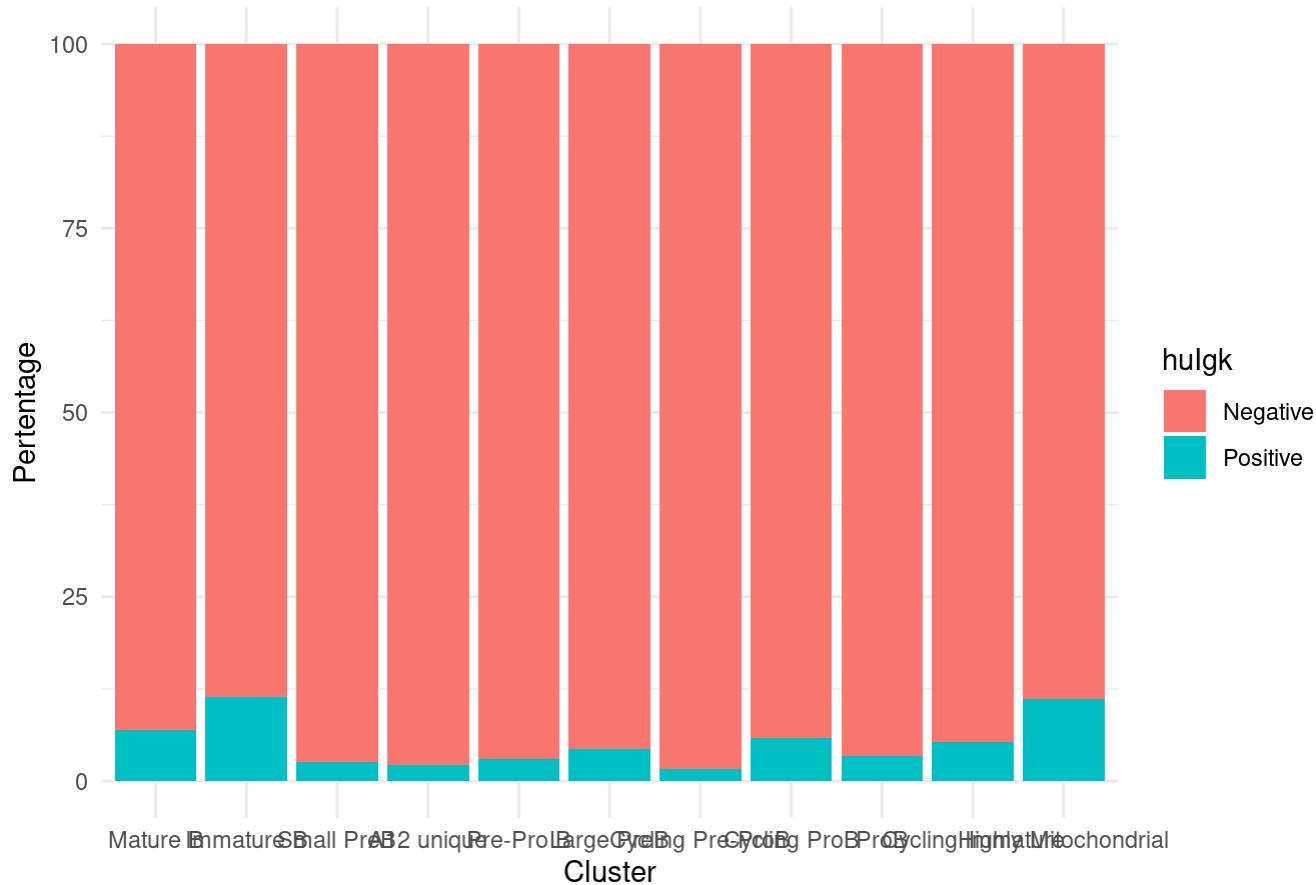
```
## `summarise()` has grouped output by 'seurat_clusters_new'. You can override
## using the `.groups` argument.
```

```
print(percentages)
```

```
### # A tibble: 22 × 4
### # Groups: seurat_clusters_new [11]
##   seurat_clusters_new huIgk_positive n percentage
##   <fct>            <chr>          <int>     <dbl>
## 1 Mature B          Negative       873     93.1
## 2 Mature B          Positive        65      6.93
## 3 Immature B        Negative      601     88.6
## 4 Immature B        Positive       77      11.4
## 5 Small PreB         Negative     1590     97.4
## 6 Small PreB         Positive       42      2.57
## 7 A12 unique         Negative     534     97.8
## 8 A12 unique         Positive       12      2.20
## 9 Pre-Pro B          Negative     323     97.0
## 10 Pre-Pro B         Positive       10      3.00
## # i 12 more rows
```

```
ggplot(percentages, aes(x = seurat_clusters_new, y = percentage, fill = huIgk_positive)) +
  geom_bar(stat = "identity", position = "stack") + # Gráfico de barras apiladas
  labs(title = "Frecuencia de células por cluster",
       x = "Cluster",
       y = "Porcentaje",
       fill = "huIgk") +
  theme_minimal()
```

Frequency of cells per cluster



Deeper investigation and comparison of A12 unique cluster versus small PreB and ProB

```
Seurat_FrE_A12 <- subset(Seurat_Object_BM_A12, seurat_clusters_new %in% c("Small PreB", "A12 unique", "ProB"))
Idents(Seurat_FrE_A12) <- Seurat_FrE_A12$simplified_clusters
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_FrE_A12, test.use = "roc", ident.1 = "A12 unique", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

Warning: The following arguments are not used: thresh.use

We print the two most relevant per cluster:

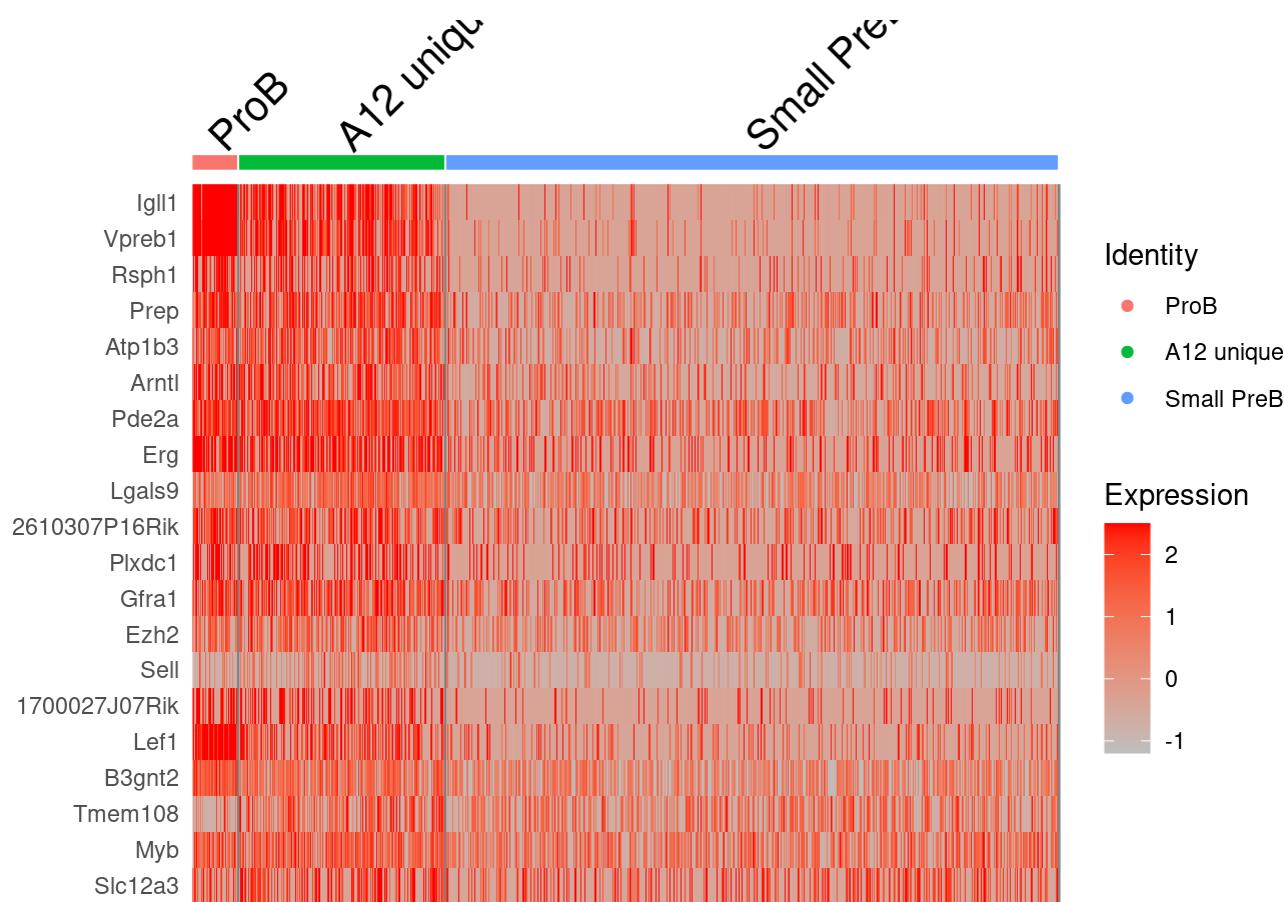
```
five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < 1) & (pct.1 > 0.25 | pct.
2 > 0.25)) %>%
  slice_max(n = 20, order_by = avg_diff) %>%
  slice_head(n = 20) %>%
  ungroup()
Idents(Seurat_FrE_A12) <- factor(
  Seurat_FrE_A12$seurat_clusters_new,
  levels = c("ProB", "Large PreB", "A12 unique", "Small PreB")
)
```

```
five_top_markers_A12_uniq_pop_BM
```

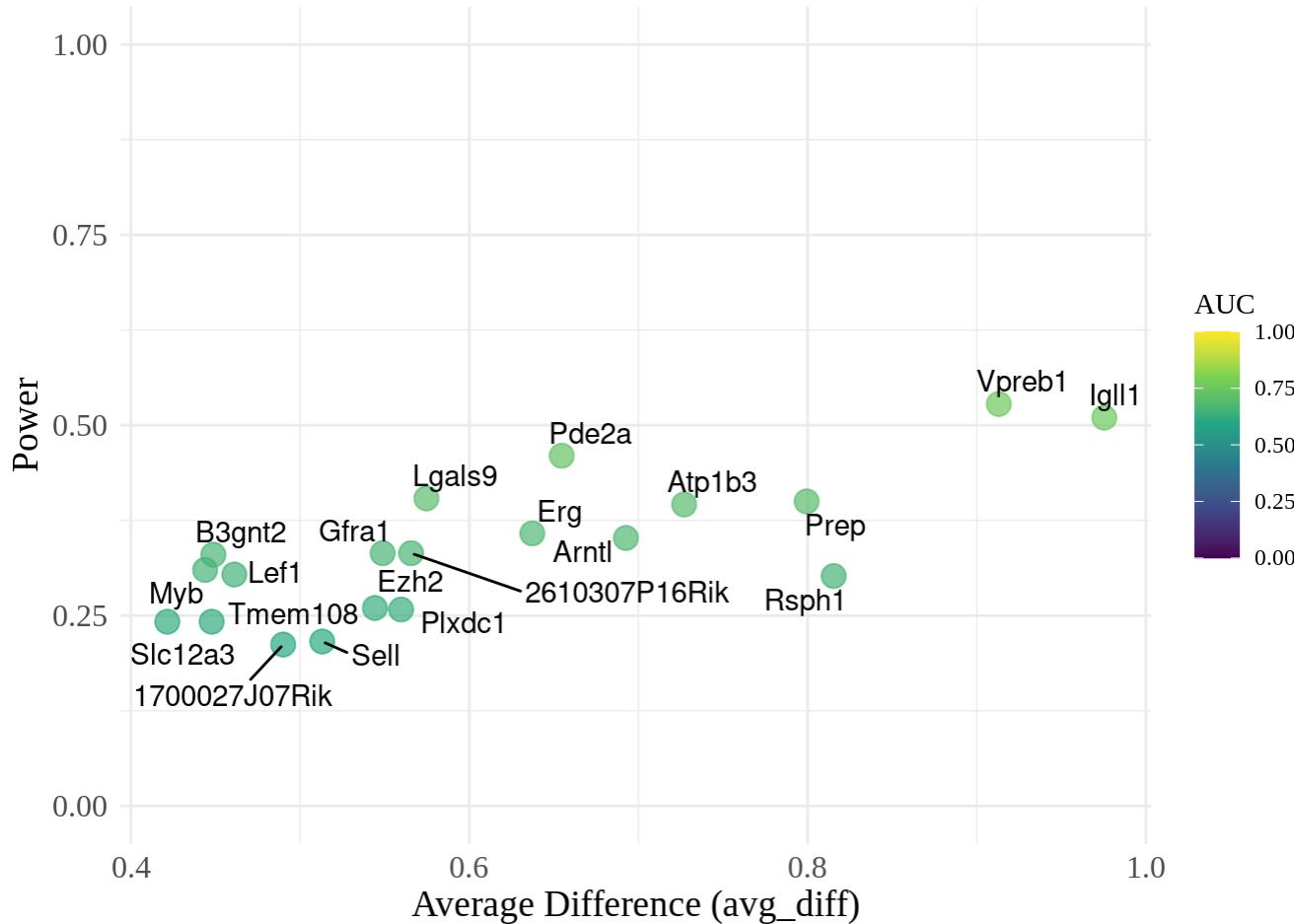
```
##          myAUC avg_diff power avg_log2FC pct.1 pct.2
## Igll1           0.755 0.9753755 0.510   1.6200732 0.674 0.154
## Vpreb1          0.764 0.9130145 0.528   1.5264787 0.709 0.179
## Rspfh1          0.651 0.8154200 0.302   1.8173674 0.456 0.171
## Prep            0.700 0.7994362 0.400   1.4698616 0.700 0.385
## Atp1b3          0.698 0.7270287 0.396   1.4255904 0.678 0.353
## Arntl            0.676 0.6926352 0.352   1.3511308 0.615 0.313
## Pde2a            0.730 0.6546684 0.460   1.1556654 0.837 0.465
## Erg              0.679 0.6372663 0.358   1.2565663 0.641 0.297
## Lgals9            0.702 0.5747151 0.404   0.9488613 0.853 0.650
## 2610307P16Rik    0.666 0.5656329 0.332   1.0988634 0.656 0.367
## Plxdc1            0.629 0.5597157 0.258   1.4073826 0.438 0.183
## Gfra1             0.666 0.5488101 0.332   1.0158136 0.709 0.437
## Ezh2              0.630 0.5442311 0.260   1.0062435 0.604 0.394
## Sell              0.608 0.5130956 0.216   1.3444049 0.381 0.175
## 1700027J07Rik    0.606 0.4899991 0.212   1.4050411 0.350 0.143
## Lef1              0.652 0.4611333 0.304   0.9687842 0.538 0.237
## B3gnt2            0.665 0.4487442 0.330   0.7475598 0.842 0.669
## Tmem108           0.621 0.4477441 0.242   0.8303607 0.647 0.451
## Myb               0.655 0.4438677 0.310   0.7674284 0.795 0.580
## Slc12a3           0.621 0.4215173 0.242   0.8819116 0.557 0.348
```

```
DoHeatmap(Seurat_FrE_A12, features =rownames(five_top_markers_A12_uniq_pop_BM) ) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



```
ggplot(five_top_markers_A12_uniq_pop_BM, aes(x = avg_diff, y = power, label = row.names(five_top_markers_A12_uniq_pop_BM))) +
  geom_point(aes(color = myAUC), alpha = 0.7, size = 4) +
  geom_text_repel() +
  scale_y_continuous(limits = c(0, 1), name = "Power") +
  scale_color_viridis_c(option = "D", name = "AUC", limits = c(0, 1)) +
  labs(
    x = "Average Difference (avg_diff)"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14)
  )
}
```



```
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_FrE_A12, test.use = "MAST", ident.1 = "A12 unique", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Warning in new_with_repaired_slots(classname = method, design = colData(sca), : Dropping illegal slot(s) thresh.use for class BayesGLMlike.
## This likely indicates a bug in an upstream package.
```

```
##
## Done!
```

```
## Combining coefficients and standard errors
```

```
## Calculating log-fold changes
```

```
## Calculating likelihood ratio tests
```

```
## Refitting on reduced model...
```

```
##
## Done!
```

```

five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < -1) & (pct.1 > 0.25 | pct.
2 > 0.25) & p_val_adj < 1e-10) %>%
  slice_min(n = 30, order_by = p_val_adj) %>%
  slice_head(n = 30) %>%
  ungroup()
Idents(Seurat_FrE_A12) <- factor(
  Seurat_FrE_A12$seurat_clusters_new,
  levels = c("Prob", "Large PreB", "A12 unique", "Small PreB")
)
five_top_markers_A12_uniq_pop_BM

```

| | p_val | avg_log2FC | pct.1 | pct.2 | p_val_adj |
|------------------|---------------|------------|-------|-------|---------------|
| ## Igll1 | 5.818799e-115 | 1.6200732 | 0.674 | 0.154 | 1.157010e-110 |
| ## Vpreb1 | 7.524862e-114 | 1.5264787 | 0.709 | 0.179 | 1.496244e-109 |
| ## Pde2a | 8.112051e-66 | 1.1556654 | 0.837 | 0.465 | 1.613000e-61 |
| ## Iglc1 | 1.245706e-61 | -3.3232483 | 0.051 | 0.386 | 2.476962e-57 |
| ## Prep | 2.627993e-56 | 1.4698616 | 0.700 | 0.385 | 5.225501e-52 |
| ## Atp1b3 | 4.211335e-56 | 1.4255904 | 0.678 | 0.353 | 8.373818e-52 |
| ## Iglc3 | 6.017085e-54 | -1.7169329 | 0.223 | 0.596 | 1.196437e-49 |
| ## Lgals9 | 1.177086e-50 | 0.9488613 | 0.853 | 0.650 | 2.340518e-46 |
| ## Erg | 2.513924e-46 | 1.2565663 | 0.641 | 0.297 | 4.998687e-42 |
| ## Cd74 | 4.920591e-46 | -2.3108590 | 0.306 | 0.595 | 9.784104e-42 |
| ## Arntl | 9.307052e-43 | 1.3511308 | 0.615 | 0.313 | 1.850614e-38 |
| ## Rsphl | 5.605685e-42 | 1.8173674 | 0.456 | 0.171 | 1.114634e-37 |
| ## Smarca4 | 1.462884e-40 | 0.6401266 | 0.967 | 0.855 | 2.908798e-36 |
| ## Lef1 | 1.925793e-37 | 0.9687842 | 0.538 | 0.237 | 3.829248e-33 |
| ## 2610307P16Rik | 3.119310e-36 | 1.0988634 | 0.656 | 0.367 | 6.202436e-32 |
| ## Gfra1 | 2.101255e-35 | 1.0158136 | 0.709 | 0.437 | 4.178136e-31 |
| ## B3gnt2 | 2.259060e-32 | 0.7475598 | 0.842 | 0.669 | 4.491914e-28 |
| ## Plxdc1 | 1.052972e-30 | 1.4073826 | 0.438 | 0.183 | 2.093729e-26 |
| ## Cd2 | 1.862888e-30 | -1.7451578 | 0.154 | 0.402 | 3.704167e-26 |
| ## Myb | 1.175146e-28 | 0.7674284 | 0.795 | 0.580 | 2.336660e-24 |
| ## Xrcc6 | 2.889422e-25 | 0.6471851 | 0.866 | 0.701 | 5.745326e-21 |
| ## Tmem163 | 7.798264e-25 | -0.9311783 | 0.416 | 0.652 | 1.550607e-20 |
| ## Ezh2 | 4.184417e-24 | 1.0062435 | 0.604 | 0.394 | 8.320294e-20 |
| ## 1700027J07Rik | 5.145382e-24 | 1.4050411 | 0.350 | 0.143 | 1.023108e-19 |
| ## Igfv1 | 2.922093e-23 | -3.2833270 | 0.112 | 0.273 | 5.810290e-19 |
| ## Ifi30 | 3.323636e-23 | -2.0006599 | 0.165 | 0.349 | 6.608719e-19 |
| ## Sell | 4.532319e-23 | 1.3444049 | 0.381 | 0.175 | 9.012062e-19 |
| ## Ccnd3 | 2.876517e-21 | 0.6542843 | 0.822 | 0.679 | 5.719667e-17 |
| ## Slc12a3 | 7.640317e-21 | 0.8819116 | 0.557 | 0.348 | 1.519201e-16 |
| ## Tmem108 | 8.121712e-20 | 0.8303607 | 0.647 | 0.451 | 1.614921e-15 |

```

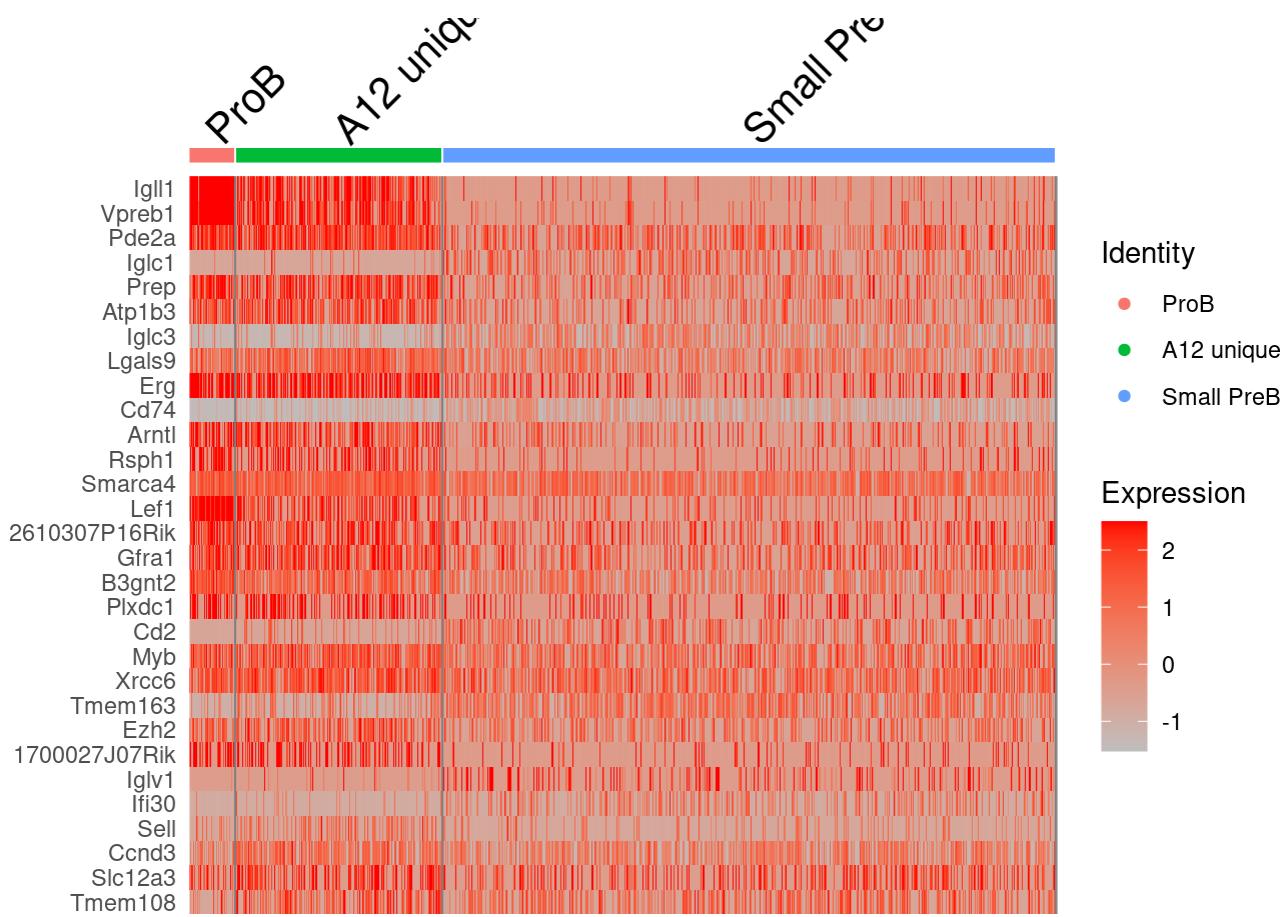
DoHeatmap(Seurat_FrE_A12, features = rownames(five_top_markers_A12_uniq_pop_BM)) +
  scale_fill_gradientn(colors = c("grey", "red"))

```

```

## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

```



```
average_expression <- AverageExpression(Seurat_FrE_A12, features = row.names(five_top_markers_A12_uniq_pop_BM), return.seurat = TRUE)
```

```
## Warning: The following 30 features were not found in the HTO assay: Igll1,
## Vpreb1, Pde2a, Iglc1, Prep, Atp1b3, Iglc3, Lgals9, Erg, Cd74, Arntl, Rsphl,
## Smarca4, Lef1, 2610307P16Rik, Gfra1, B3gnt2, Plxdc1, Cd2, Myb, Xrcc6, Tmem163,
## Ezh2, 1700027J07Rik, Igfv1, Ifi30, Sell, Ccnd3, Slc12a3, Tmem108
```

```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

```
genes_of_interest <- row.names(average_expression@assays$RNA$scale.data)

expr_subset <- average_expression@assays$RNA$scale.data[genes_of_interest, ]

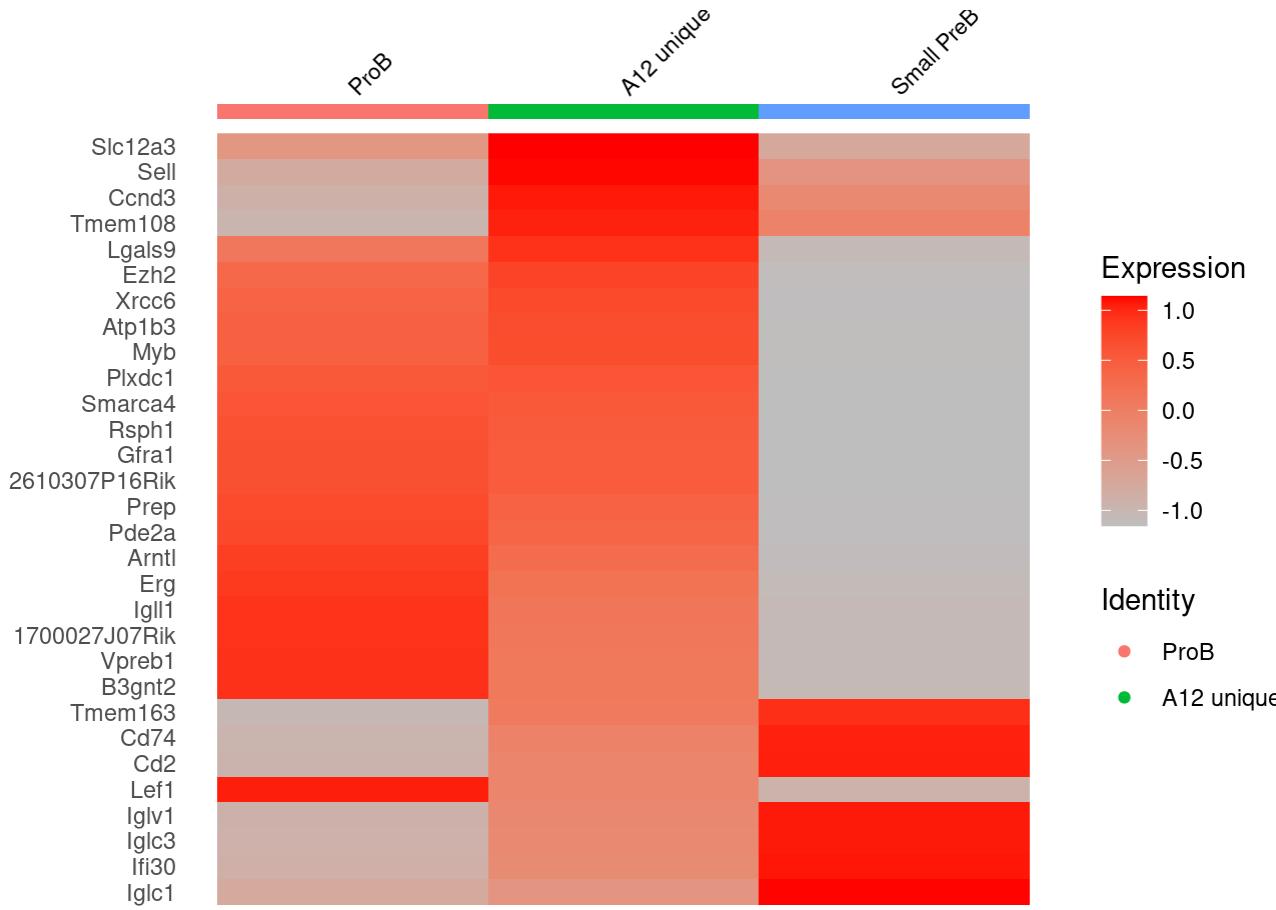
ordered_genes <- rownames(expr_subset[order(expr_subset[, "A12 unique"], decreasing = TRUE), ])

DoHeatmap(average_expression, features = ordered_genes, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
```

```
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

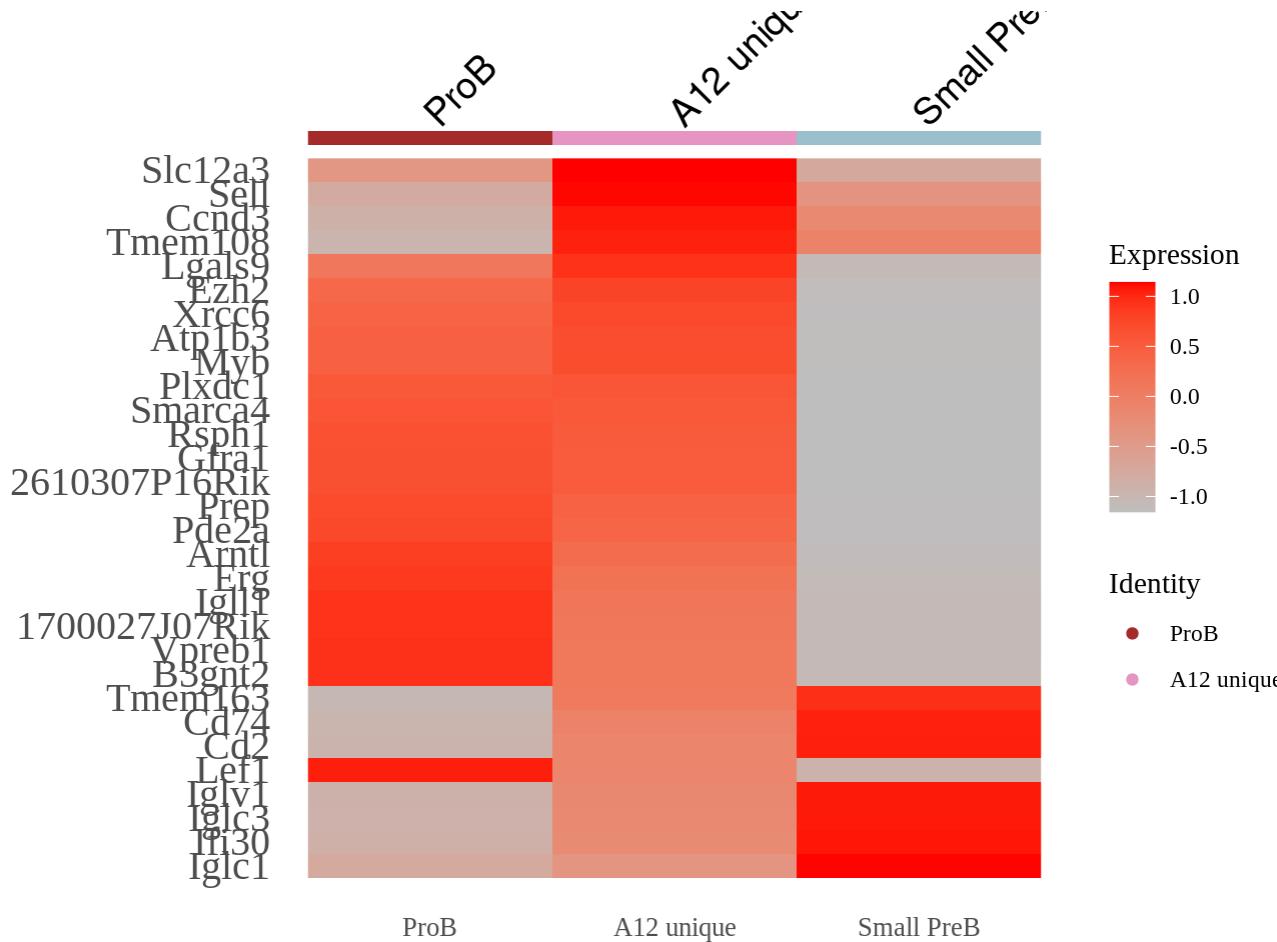


```
DoHeatmap(average_expression, features = ordered_genes, group.colors = c(
  "ProB" = "#A52A2A",
  "Small PreB" = "lightblue3",
  "A12 unique" = "#E694C1",
  "Large PreB" = "#A694C1"
), size = 5, draw.lines = FALSE) +
scale_fill_gradientn(colors = c("grey", "red")) +
theme(
  text = element_text(family = "Times New Roman"),
  axis.text.x = element_text(size = 10, family = "Times New Roman"),
  axis.text.y = element_text(size = 15, family = "Times New Roman"),
  axis.title = element_text(size = 12, family = "Times New Roman"),
  plot.title = element_text(size = 14, face = "bold", family = "Times New Roman")
)
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
```

```
## (`geom_point()`).
```



```
average_expression <- AverageExpression(Seurat_FrE_A12, features = row.names(five_top_markers_A12_uniq_pop_BM))$RNA %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  pivot_longer(cols = -gene, names_to = "cluster", values_to = "expression")
```

```
## Warning: The following 30 features were not found in the HTO assay: Igll1,
## Vpreb1, Pde2a, Igclc1, Prep, Atplb3, Igllc3, Lgals9, Erg, Cd74, Arntl, Rspn1,
## Smarca4, Lef1, 2610307P16Rik, Gfral, B3gnt2, Plxdc1, Cd2, Myb, Xrcc6, Tmem163,
## Ezh2, 1700027J07Rik, Igvl1, Ifi30, Sell, Ccnd3, Slc12a3, Tmem108
## Warning: None of the features specified were found in the HTO assay.
```

```
five_top_markers_A12_uniq_pop_BM <- five_top_markers_A12_uniq_pop_BM %>%
  dplyr::mutate(gene = rownames(five_top_markers_A12_uniq_pop_BM))

data_for_dotplot <- five_top_markers_A12_uniq_pop_BM %>%
  dplyr::select(gene, avg_log2FC, p_val_adj) %>%
  left_join(average_expression, by = "gene") %>%
  mutate(
    p_val_adj = ifelse(p_val_adj == 0, 1e-300, p_val_adj),
    significance = -log10(p_val_adj)
  )
```

```
data_for_dotplot <- data_for_dotplot %>%
  mutate(expression_scaled = log1p(expression)) #  $\log1p(x) = \log(1 + x)$ 

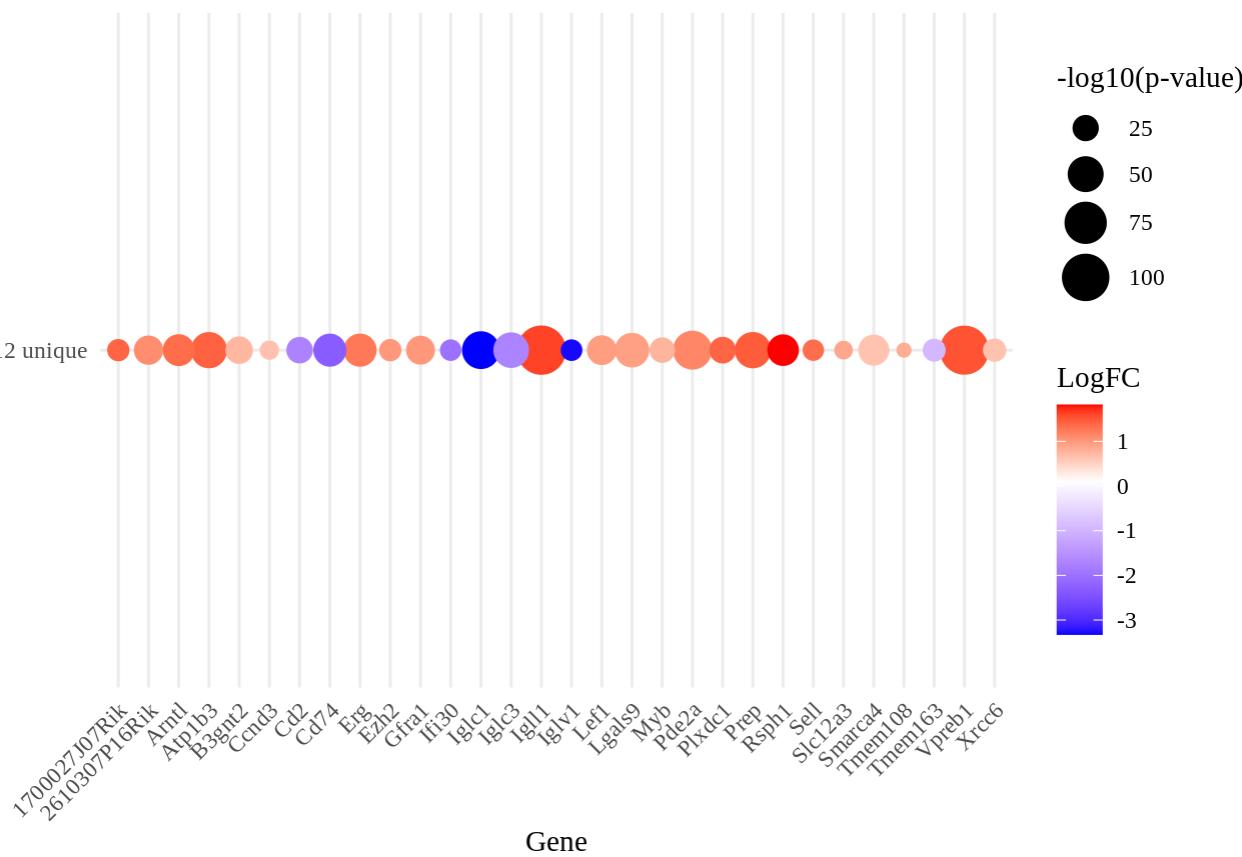
data_for_dotplot_A12 <- data_for_dotplot %>%
  filter(cluster == "A12 unique")

ggplot(data_for_dotplot_A12, aes(x = gene, y = cluster)) +
  geom_point(aes(size = significance, color = avg_log2FC)) +
  scale_color_gradientn(
    colors = c("blue", "white", "red"),
    values = scales::rescale(c(-3, 0, 1.5)),
    name = "LogFC"
  ) +
  scale_size_continuous(range = c(2, 8), name = "-log10(p-value)") +
  labs(
    x = "Gene",
    y = "Cluster A12", # Etiqueta del eje Y solo para A12
    title = "Dot plot of gene expression and significance for cluster A12",
    subtitle = "Size indicates -log10(p-value); Color indicates logFC"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, family = "Times New Roman"),
    text = element_text(family = "Times New Roman"),
    legend.position = "right"
  )
```

Dot plot of gene expression and significance for cluster A12

Size indicates $-\log_{10}(p\text{-value})$; Color indicates logFC

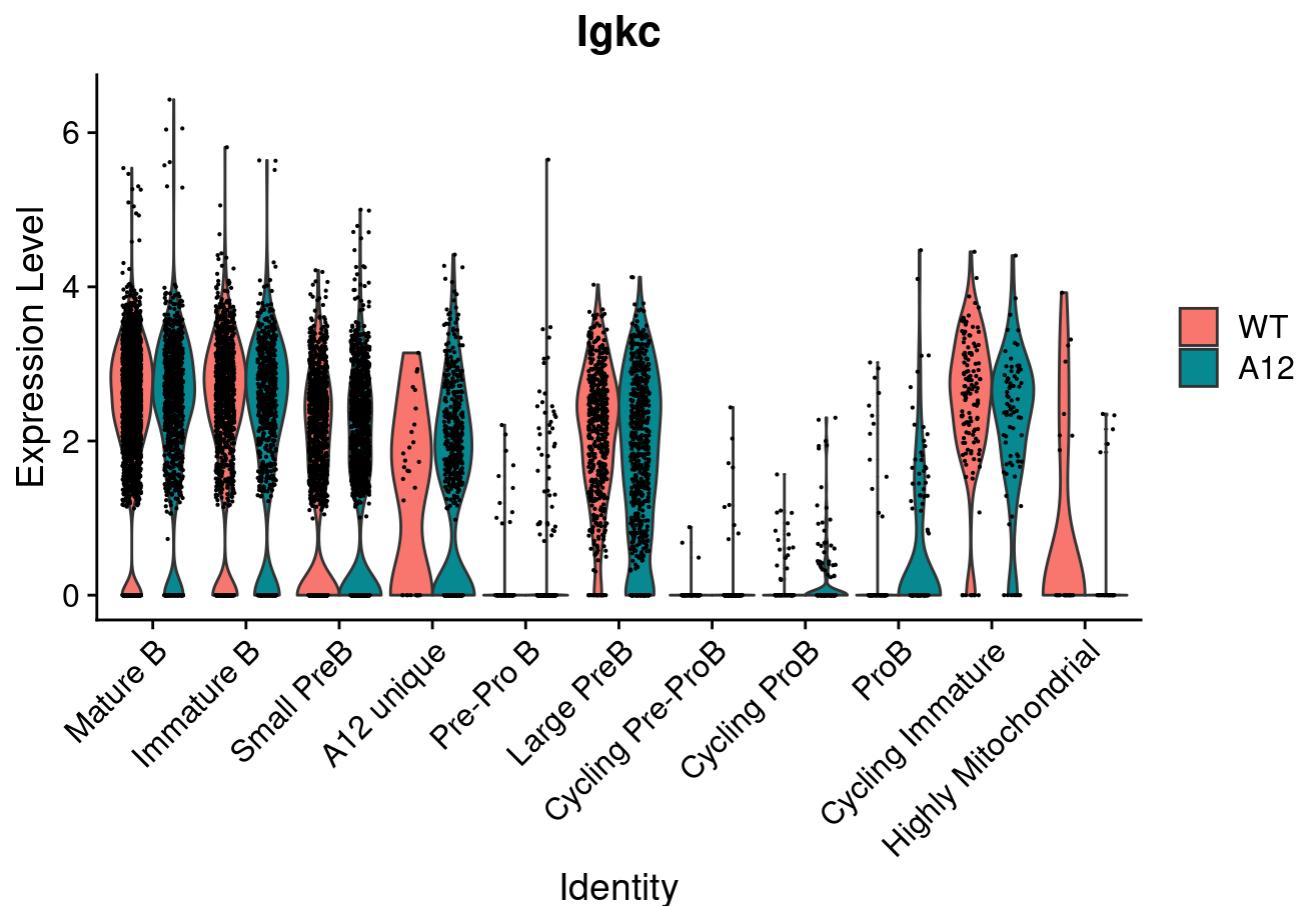
Cluster A12



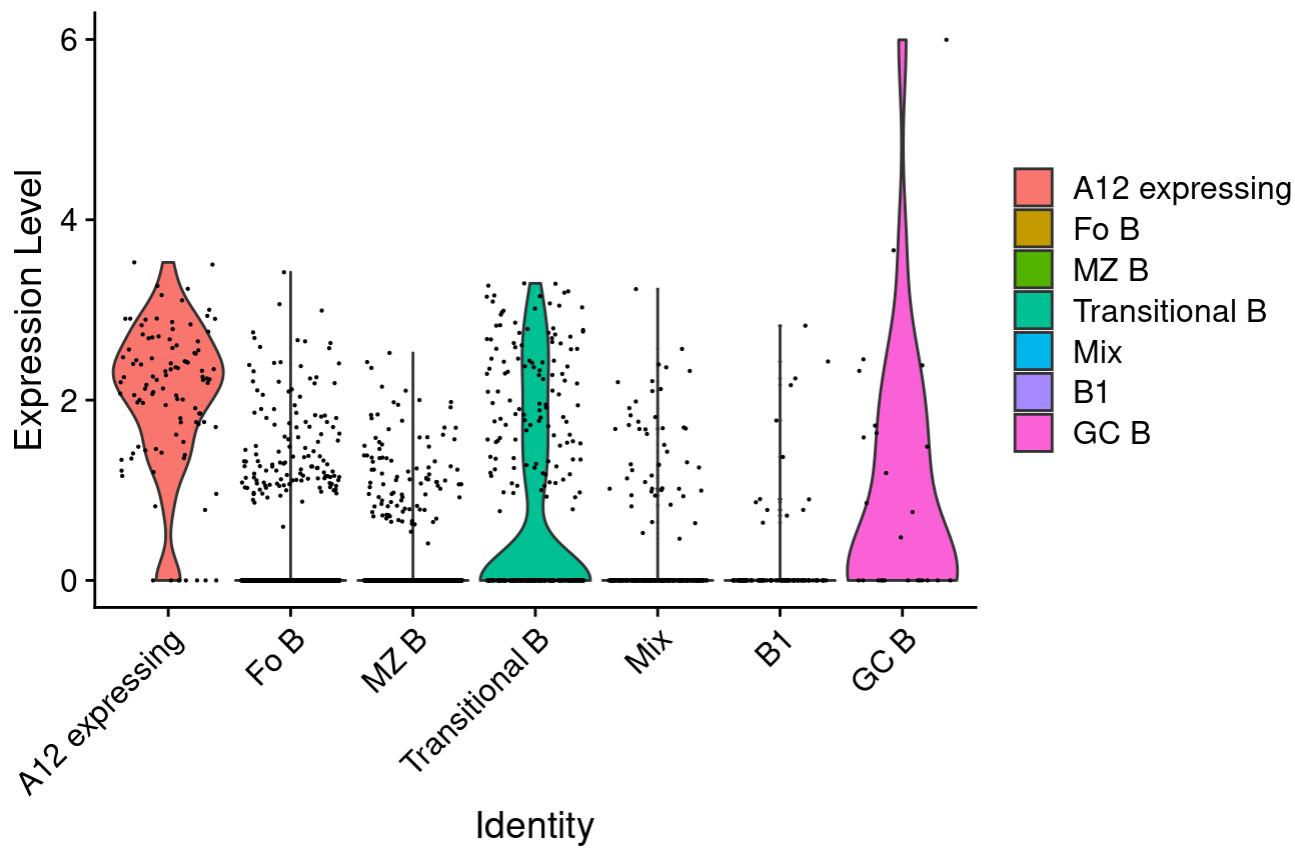
A12 expression Violin

```
VlnPlot(Seurat_Object_BM_selected_Bcells_idents, features = "Igkc", group.by = "seurat_clusters_new", split.by = "genotype")
```

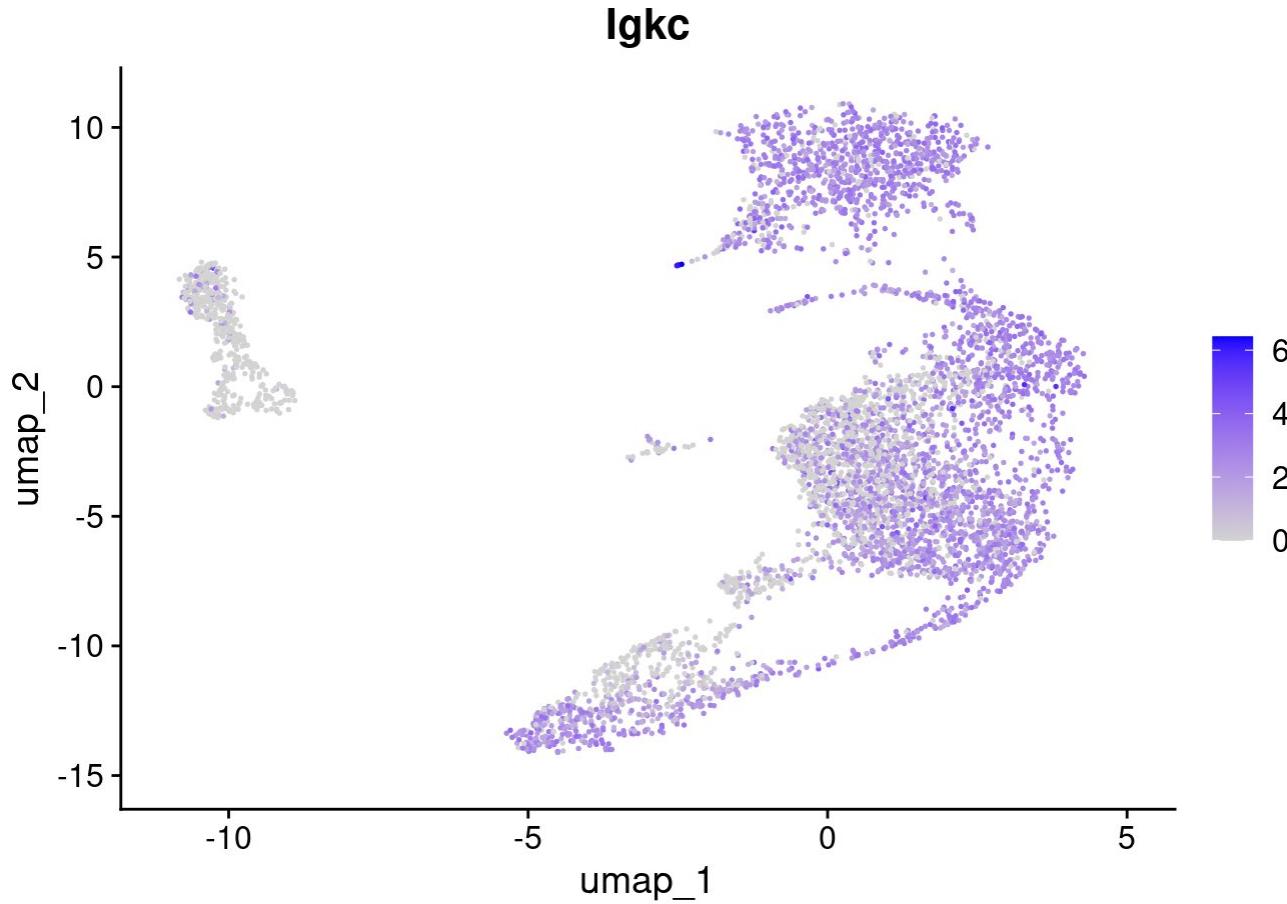
```
## The default behaviour of split.by has changed.
## Separate violin plots are now plotted side-by-side.
## To restore the old behaviour of a single split violin,
## set split.plot = TRUE.
##
## This message will be shown once per session.
```



```
VlnPlot(Seurat_Object_SP_A12, features = "A12", group.by = "seurat_clusters_new_fin")
```

A12

```
FeaturePlot(Seurat_Object_BM_A12, features = "IgkC")
```



A12 expresion Barplot

```

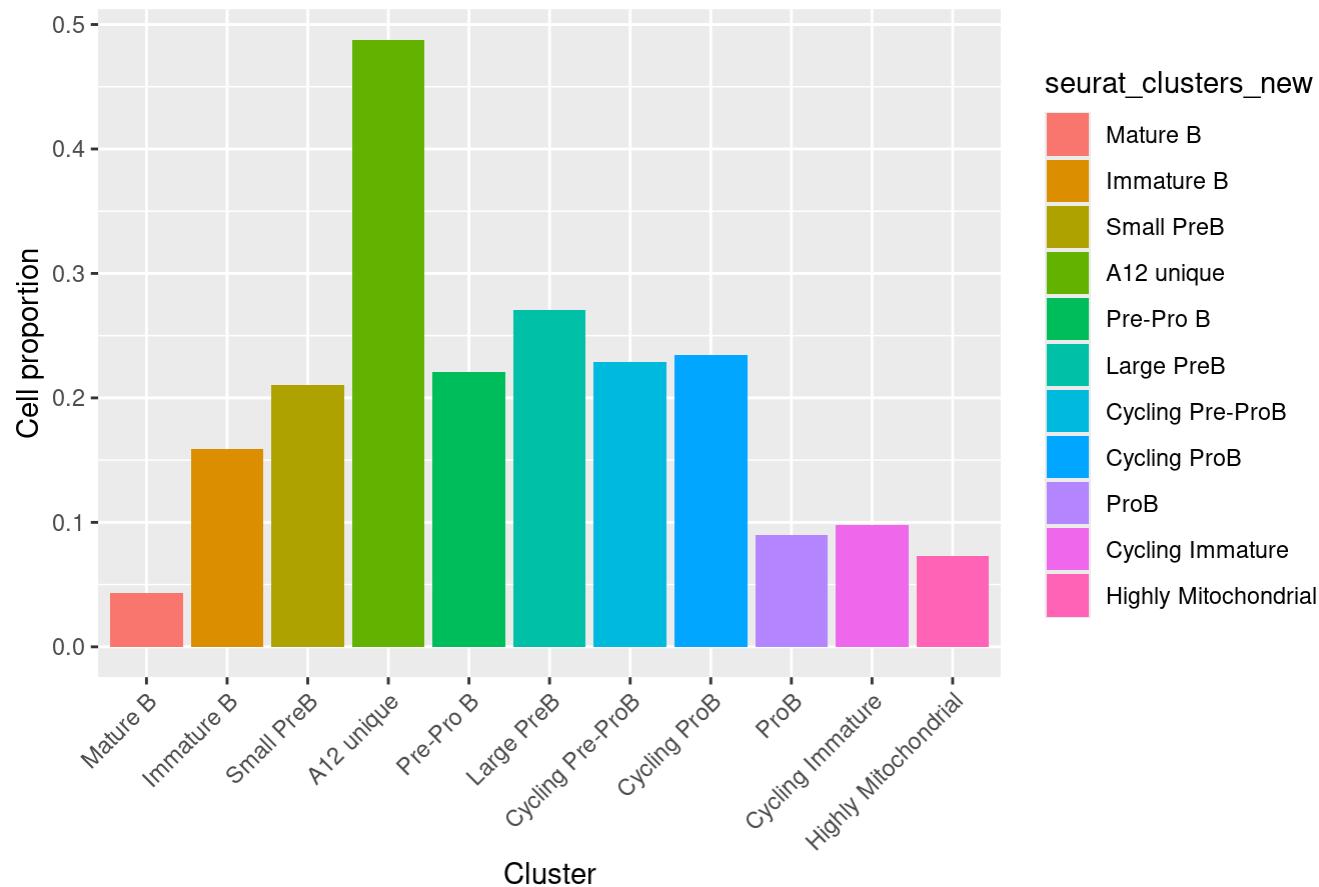
data <- Seurat_Object_BM_selected_Bcells_idents@meta.data
desired_order <- c("Pre-Pro B", "Cycling Pre-ProB", "ProB", "Cycling_ProB",
                  "Large PreB", "A12 unique", "Small PreB", "Immature B",
                  "Cycling Immature", "Mature B", "High Mitochondrial", "NA")

exprs <- Seurat_Object_BM_selected_Bcells_idents@assays$RNA$counts["A12",]
data$seurat_clusters <- factor(data$seurat_clusters_new, levels = desired_order)
data$Gene_A12_Expression <- exprs > 0

prop_data <- data %>%
  group_by(seurat_clusters_new) %>%
  summarise(Proportion = mean(Gene_A12_Expression))

ggplot(prop_data, aes(x = seurat_clusters_new, y = Proportion, fill = seurat_clusters_new)) +
  geom_bar(stat = "identity") +
  labs(title = "A12 expressing cell proportion",
       x = "Cluster",
       y = "Cell proportion") + theme(axis.text.x = element_text(angle = 45, hjust = 1))
  
```

A12 expressing cell proportion



Bone Marrow Statistics

```
# Obtain percentages for each cluster in each mouse
data_summary <- Seurat_Object_BM_selected_Bcells_idents@meta.data %>%
  group_by(genotype, mice, simplified_clusters) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
```

```
## `summarise()` has grouped output by 'genotype', 'mice'. You can override using
## the ` `.groups` argument.
```

```
# Filter out unwanted clusters
data_summary <- data_summary %>%
  filter(!is.na(simplified_clusters) & simplified_clusters != "Highly Mitochondrial")

# Create empty lists to store percentages for each genotype
A12_test <- list()
WT_test <- list()
for (i in unique(data_summary$simplified_clusters)) {
  A12_test[[as.character(i)]] <- data_summary[data_summary$genotype == "A12" & data_summary$simplified_clusters == i, ]
  WT_test[[as.character(i)]] <- data_summary[data_summary$genotype == "WT" & data_summary$simplified_clusters == i, ]
```

```

}

# Create an empty list to store t-test results
t_test_results <- list()

# Perform t-tests for each cluster between genotypes
for (i in names(A12_test)) {
  group1 <- A12_test[[i]]$percentage
  group2 <- WT_test[[i]]$percentage

  # Perform the t-test
  result <- t.test(group1, group2)

  # Store the results in the list
  t_test_results[[paste("Comparison", i)]] <- result
}

# Rename the result names to simplify
names(t_test_results) <- sub("Comparison ", "", names(t_test_results))

# Extract p-values from the t-test results
p_values <- sapply(t_test_results, function(res) res$p.value)
p_values

```

```

##      Pre-Pro B          ProB    Large PreB     A12 unique    Small PreB    Immature B
## 0.0007310232 0.0864230292 0.3157475506 0.0094213740 0.1824642825 0.0572673531
##      Mature B
## 0.0743260305

```

```

# Calculate log fold change and add significance markers
mean_per_cluster_A12 <- sapply(A12_test, function(df) mean(df$percentage, na.rm = TRUE))
mean_per_cluster_WT <- sapply(WT_test, function(df) mean(df$percentage, na.rm = TRUE))

# Extract p-values
p_value_list <- unlist(lapply(t_test_results, function(res) res$p.value))

# Calculate Log Fold Change (LFC)
log_fold_change <- log2(mean_per_cluster_A12 / mean_per_cluster_WT)

# Create a data frame with the results
results_df <- data.frame(
  cluster = names(A12_test),
  mean_A12 = mean_per_cluster_A12,
  mean_WT = mean_per_cluster_WT,
  log_fold_change = log_fold_change,
  p_value = p_value_list
)

# Determine levels of significance or include the p-value if between 0.05 and 0.1
results_df$significance <- ifelse(
  results_df$p_value <= 0.001, "***",
  ifelse(

```

```

results_df$p_value <= 0.01, "***",
ifelse(
  results_df$p_value <= 0.05, "**",
  ifelse(
    results_df$p_value <= 0.1,
    sprintf("%.3f", results_df$p_value), # Format to display p-value with 3 decimals
    ""
  )
)
)
)
)

print(results_df)

```

| | cluster | mean_A12 | mean_WT | log_fold_change | p_value |
|---------------|--------------|-----------|------------|-----------------|--------------|
| ## Pre-Pro B | Pre-Pro B | 9.688210 | 3.3749369 | 1.5213696 | 0.0007310232 |
| ## ProB | ProB | 4.853409 | 2.8169965 | 0.7848407 | 0.0864230292 |
| ## Large PreB | Large PreB | 11.859994 | 8.5421559 | 0.4734312 | 0.3157475506 |
| ## A12 unique | A12 unique | 10.311904 | 0.5367323 | 4.2639644 | 0.0094213740 |
| ## Small PreB | Small PreB | 30.818131 | 24.7363919 | 0.3171443 | 0.1824642825 |
| ## Immature B | Immature B | 14.240066 | 20.1547302 | -0.5011626 | 0.0572673531 |
| ## Mature B | Mature B | 17.718389 | 39.3726155 | -1.1519451 | 0.0743260305 |
| ## | significance | | | | |
| ## Pre-Pro B | | *** | | | |
| ## ProB | | 0.086 | | | |
| ## Large PreB | | ** | | | |
| ## A12 unique | | | | | |
| ## Small PreB | | | | | |
| ## Immature B | | 0.057 | | | |
| ## Mature B | | 0.074 | | | |

```

# Add significance levels to the summary data for use in the plot
data_summary <- data_summary %>%
  left_join(results_df %>% select(cluster, significance), by = c("simplified_clusters" = "cluster"))

# Define the order of clusters for the plot
cluster_order <- c("Pre-Pro B", "Cycling Pre-ProB", "ProB", "Cycling ProB",
                    "Large PreB", "A12 unique", "Small PreB", "Immature B",
                    "Cycling Immature", "Mature B")

data_summary$simplified_clusters <- factor(data_summary$simplified_clusters, levels = cluster_order)

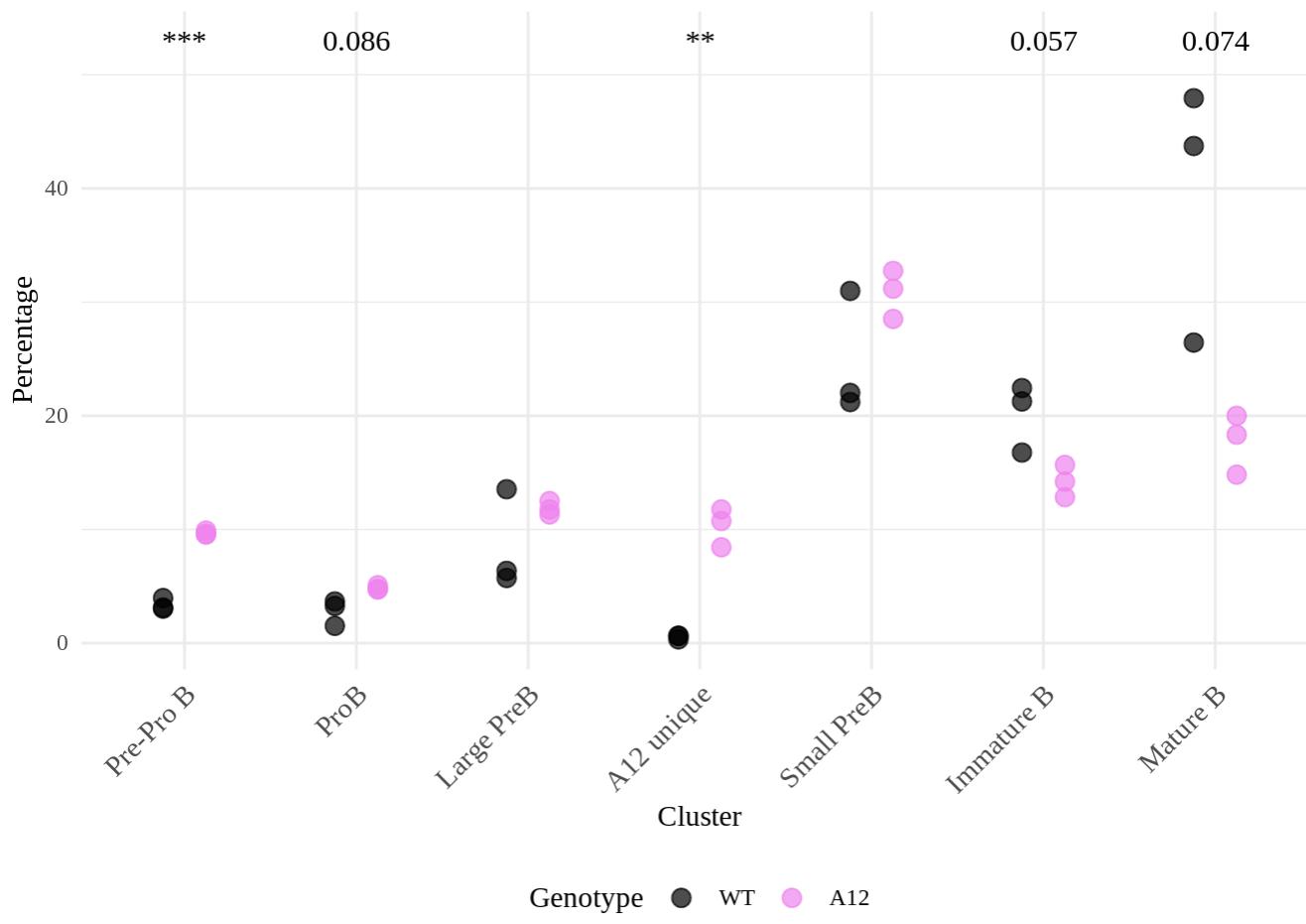
# Create the plot with log fold change and significance (asterisks or p-values)
ggplot(data_summary, aes(x = simplified_clusters, y = percentage, color = genotype)) +
  geom_point(size = 3, position = position_dodge(width = 0.5), alpha = 0.7) +
  geom_text(
    data = results_df,
    aes(x = cluster, y = max(data_summary$percentage) + 5, label = significance),
    inherit.aes = FALSE,
    size = 4,
    family = "Times New Roman"
  )

```

```

) +
scale_color_manual(values = c("WT" = "black", "A12" = "violet")) +
labs(
  x = "Cluster",
  y = "Percentage",
  color = "Genotype"
) +
theme_minimal() +
theme(
  text = element_text(family = "Times New Roman"),
  axis.text.x = element_text(size = 11, angle = 45, hjust = 1, family = "Times New Roman"),
  legend.position = "bottom"
)

```



Saving work

```

#saveRDS(Seurat_Object_BM_selected_Bcells_ids, file = ".../Results/BM_final_Seurat_TFM_simp.RDS")
#saveRDS(Seurat_Object_SP_selected_Bcells_ids, file = ".../Results/SP_final_Seurat_TFM_simp.RDS")
#Seurat_Object_BM_selected_Bcells_ids <- LoadSeuratRds("../Results/BM_final_Seurat_TFM_simp.RDS")
#Seurat_Object_SP_selected_Bcells_ids <- LoadSeuratRds("../Results/SP_final_Seurat_TFM_simp.RDS")

```

Explore Mature clusters in bone marrow, focusing in cluster 7 related with autoimmunity

```
Seurat_Mature_A12 <- subset(Seurat_Object_BM_A12, seurat_clusters %in% c("7", "0", "15", "16"))
)
Idents(Seurat_Mature_A12) <- Seurat_Mature_A12$seurat_clusters
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_Mature_A12, test.use = "roc", ident
.1 = "7", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

Warning: The following arguments are not used: thresh.use

We print the two most relevant per cluster:

```
five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < 1) & (pct.1 > 0.25 | pct.
2 > 0.25)) %>%
  slice_max(n = 20, order_by = power) %>%
  slice_head(n = 20) %>%
  ungroup()

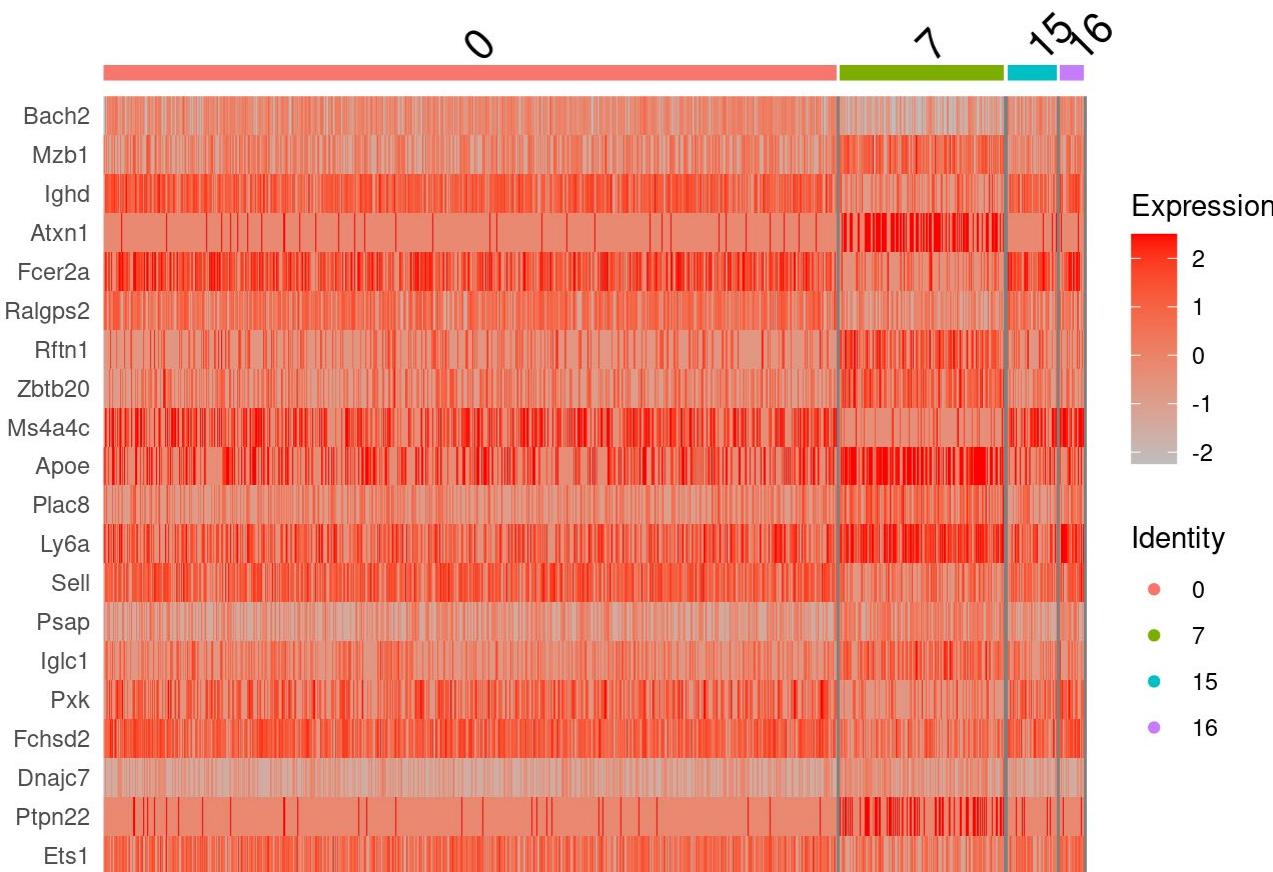
five_top_markers_A12_uniq_pop_BM
```

| | myAUC | avg_diff | power | avg_log2FC | pct.1 | pct.2 |
|------------|-------|------------|-------|------------|-------|-------|
| ## Bach2 | 0.213 | -0.8053844 | 0.574 | -1.3480546 | 0.698 | 0.922 |
| ## Mzbl1 | 0.775 | 1.0669444 | 0.550 | 1.7849321 | 0.893 | 0.624 |
| ## Ighd | 0.232 | -0.7754079 | 0.536 | -1.3617565 | 0.465 | 0.852 |
| ## Atxn1 | 0.767 | 0.9961836 | 0.534 | 3.6637883 | 0.585 | 0.046 |
| ## Fcer2a | 0.241 | -0.9262404 | 0.518 | -1.8426458 | 0.314 | 0.743 |
| ## Ralgps2 | 0.264 | -0.6889658 | 0.472 | -1.2087121 | 0.667 | 0.842 |
| ## Rftn1 | 0.732 | 0.7966093 | 0.464 | 1.7280106 | 0.711 | 0.277 |
| ## Zbtb20 | 0.726 | 0.8240384 | 0.452 | 1.5229153 | 0.811 | 0.457 |
| ## Ms4a4c | 0.279 | -0.9420313 | 0.442 | -2.4072188 | 0.182 | 0.571 |
| ## Apoe | 0.717 | 1.0608696 | 0.434 | 1.8431878 | 0.742 | 0.412 |
| ## Plac8 | 0.715 | 0.9427127 | 0.430 | 1.6694035 | 0.767 | 0.444 |
| ## Ly6a | 0.707 | 0.7164554 | 0.414 | 1.1656672 | 0.874 | 0.584 |
| ## Sell | 0.299 | -0.6055311 | 0.402 | -1.1128415 | 0.528 | 0.775 |
| ## Psap | 0.700 | 0.5589700 | 0.400 | 1.0461952 | 0.818 | 0.502 |
| ## Igldc1 | 0.696 | 0.8871103 | 0.392 | 1.5984088 | 0.730 | 0.421 |
| ## Pxk | 0.318 | -0.6401232 | 0.364 | -1.5196494 | 0.283 | 0.570 |
| ## Fchsd2 | 0.319 | -0.5097996 | 0.362 | -0.8662310 | 0.673 | 0.852 |
| ## Dnajc7 | 0.679 | 0.6716188 | 0.358 | 1.2751320 | 0.780 | 0.445 |
| ## Ptpn22 | 0.678 | 0.7029096 | 0.356 | 3.2767422 | 0.403 | 0.044 |
| ## Ets1 | 0.322 | -0.4865654 | 0.356 | -0.8732968 | 0.648 | 0.807 |

```
DoHeatmap(Seurat_Mature_A12, features =rownames(five_top_markers_A12_uniq_pop_BM)) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

Scale for fill is already present.

```
## Adding another scale for fill, which will replace the existing scale.
```



```
average_expression <- AverageExpression(Seurat_Mature_A12, features = row.names(five_top_markers_A12_uniq_pop_BM), return.seurat = TRUE)
```

```
## Warning: The following 20 features were not found in the HTO assay: Bach2,
## Mzb1, Ighd, Atxn1, Fcer2a, Ralgps2, Rftn1, Zbtb20, Ms4a4c, Apoe, Plac8, Ly6a,
## Sell, Psap, Iglc1, Pxk, Fchsd2, Dnajc7, Ptpn22, Ets1
```

```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

```
genes_of_interest <- row.names(average_expression@assays$RNA$scale.data)

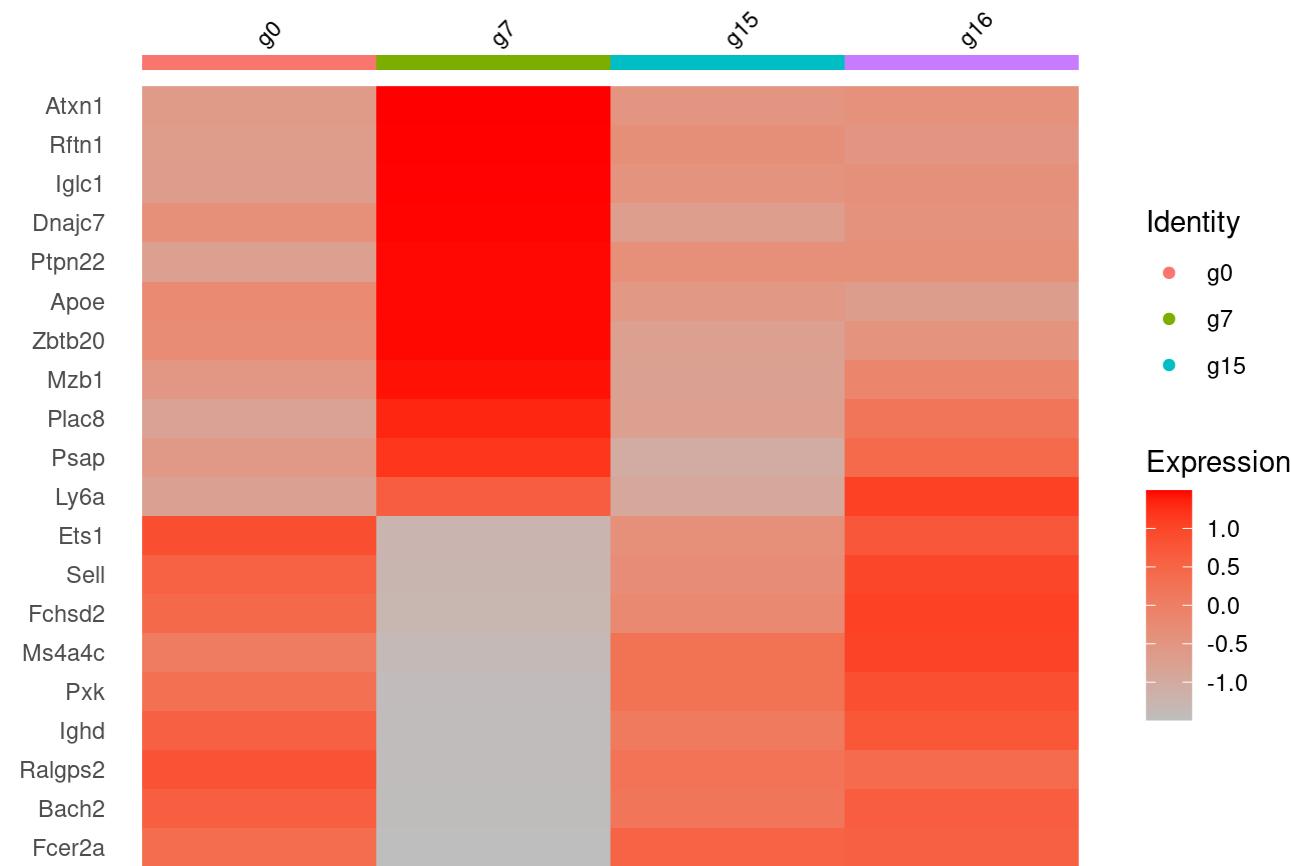
expr_subset <- average_expression@assays$RNA$scale.data[genes_of_interest, ]

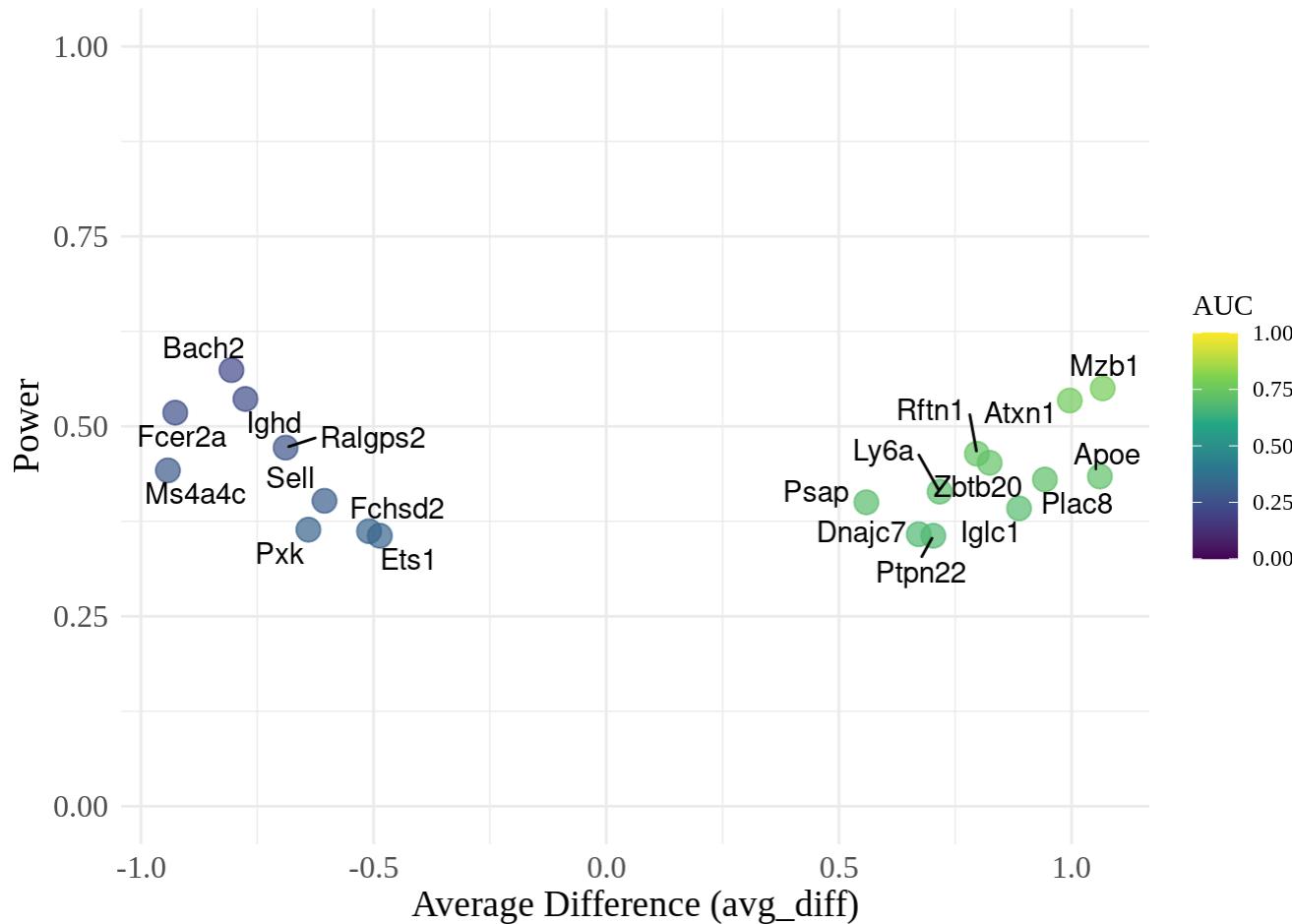
ordered_genes <- rownames(expr_subset[order(expr_subset[, "g7"], decreasing = TRUE), ])

DoHeatmap(average_expression, features = ordered_genes, size = 3, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_point()`).
```





```
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_Mature_A12, test.use = "MAST", iden.t.i = "7", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Warning in new_with_repaired_slots(classname = method, design = colData(sca), : Dropping illegal slot(s) thresh.use for class BayesGLMlike.
## This likely indicates a bug in an upstream package.
```

```
##
## Done!
```

```
## Combining coefficients and standard errors
```

```
## Calculating log-fold changes
```

```
## Calculating likelihood ratio tests
```

```
## Refitting on reduced model...
```

```
##
## Done!
```

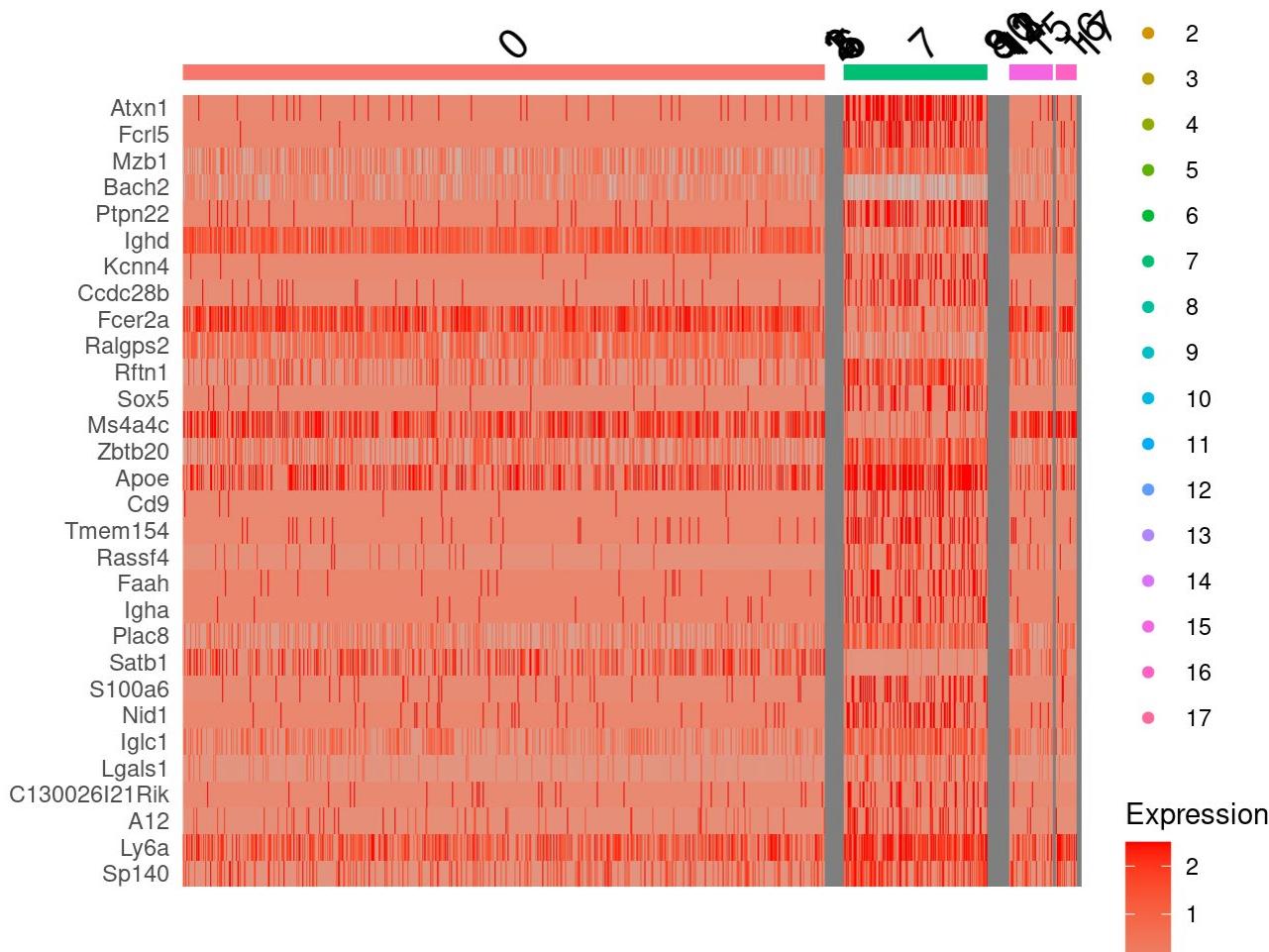
```
five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < -1) & (pct.1 > 0.25 | pct.
2 > 0.25) & p_val_adj < 1e-10) %>%
  slice_min(n = 30, order_by = p_val_adj) %>%
  slice_head(n = 30) %>%
  ungroup()
```

five_top_markers_A12_uniq_pop_BM

| | p_val | avg_log2FC | pct.1 | pct.2 | p_val_adj |
|------------------|--------------|------------|-------|-------|--------------|
| ## Atxn1 | 8.500519e-54 | 3.663788 | 0.585 | 0.046 | 1.690243e-49 |
| ## Fcrl5 | 5.322084e-39 | 5.095527 | 0.346 | 0.008 | 1.058243e-34 |
| ## Mzb1 | 3.322292e-37 | 1.784932 | 0.893 | 0.624 | 6.606046e-33 |
| ## Bach2 | 2.595808e-35 | -1.348055 | 0.698 | 0.922 | 5.161505e-31 |
| ## Ptpn22 | 6.313900e-30 | 3.276742 | 0.403 | 0.044 | 1.255456e-25 |
| ## Ighd | 1.629313e-29 | -1.361757 | 0.465 | 0.852 | 3.239725e-25 |
| ## Kcnn4 | 2.083951e-29 | 4.284617 | 0.283 | 0.010 | 4.143728e-25 |
| ## Ccdc28b | 6.258935e-29 | 2.877635 | 0.377 | 0.041 | 1.244527e-24 |
| ## Fcer2a | 1.455099e-27 | -1.842646 | 0.314 | 0.743 | 2.893319e-23 |
| ## Ralgps2 | 1.210293e-25 | -1.208712 | 0.667 | 0.842 | 2.406546e-21 |
| ## Rftn1 | 4.139785e-25 | 1.728011 | 0.711 | 0.277 | 8.231549e-21 |
| ## Sox5 | 7.844637e-24 | 4.146110 | 0.270 | 0.017 | 1.559828e-19 |
| ## Ms4a4c | 8.697841e-24 | -2.407219 | 0.182 | 0.571 | 1.729479e-19 |
| ## Zbtb20 | 1.704250e-23 | 1.522915 | 0.811 | 0.457 | 3.388730e-19 |
| ## Apoe | 4.339468e-23 | 1.843188 | 0.742 | 0.412 | 8.628598e-19 |
| ## Cd9 | 5.466252e-23 | 3.867121 | 0.252 | 0.014 | 1.086910e-18 |
| ## Tmem154 | 5.792753e-23 | 2.573199 | 0.333 | 0.045 | 1.151831e-18 |
| ## Rassf4 | 6.050239e-23 | 3.072713 | 0.346 | 0.045 | 1.203029e-18 |
| ## Faah | 1.601412e-22 | 3.046591 | 0.270 | 0.023 | 3.184248e-18 |
| ## Igaha | 2.801673e-22 | 8.225326 | 0.264 | 0.022 | 5.570846e-18 |
| ## Plac8 | 3.187407e-22 | 1.669404 | 0.767 | 0.444 | 6.337840e-18 |
| ## Satb1 | 1.136064e-21 | -3.594431 | 0.057 | 0.390 | 2.258949e-17 |
| ## S100a6 | 1.159881e-20 | 3.932078 | 0.314 | 0.046 | 2.306308e-16 |
| ## Nid1 | 8.273811e-20 | 2.801716 | 0.302 | 0.040 | 1.645165e-15 |
| ## Iglc1 | 2.525234e-19 | 1.598409 | 0.730 | 0.421 | 5.021175e-15 |
| ## Lgals1 | 9.570988e-19 | 3.166324 | 0.371 | 0.086 | 1.903095e-14 |
| ## C130026I21Rik | 3.667351e-18 | 2.062972 | 0.252 | 0.039 | 7.292160e-14 |
| ## A12 | 8.676903e-18 | 4.296793 | 0.346 | 0.073 | 1.725315e-13 |
| ## Ly6a | 4.585993e-17 | 1.165667 | 0.874 | 0.584 | 9.118789e-13 |
| ## Sp140 | 5.075103e-17 | 1.296655 | 0.629 | 0.266 | 1.009133e-12 |

```
DoHeatmap(Seurat_Mature_A12, features =rownames(five_top_markers_A12_uniq_pop_BM), group.by =
"seurat_clusters") +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



```
average_expression <- AverageExpression(Seurat_Mature_A12, features = row.names(five_top_markers_A12_uniq_pop_BM), return.seurat = TRUE, group.by = "seurat_clusters")
```

```
## Warning: The following 30 features were not found in the HTO assay: Atxn1,
## Fcrl5, Mzb1, Bach2, Ptpn22, Ighd, Kcn4, Ccdc28b, Fcer2a, Ralgps2, Rftn1, Sox5,
## Ms4a4c, Zbtb20, Apoe, Cd9, Tmem154, Rassf4, Faah, Igaha, Plac8, Satb1, S100a6,
## Nid1, Iglc1, Lgals1, C130026I21Rik, A12, Ly6a, Sp140
```

```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

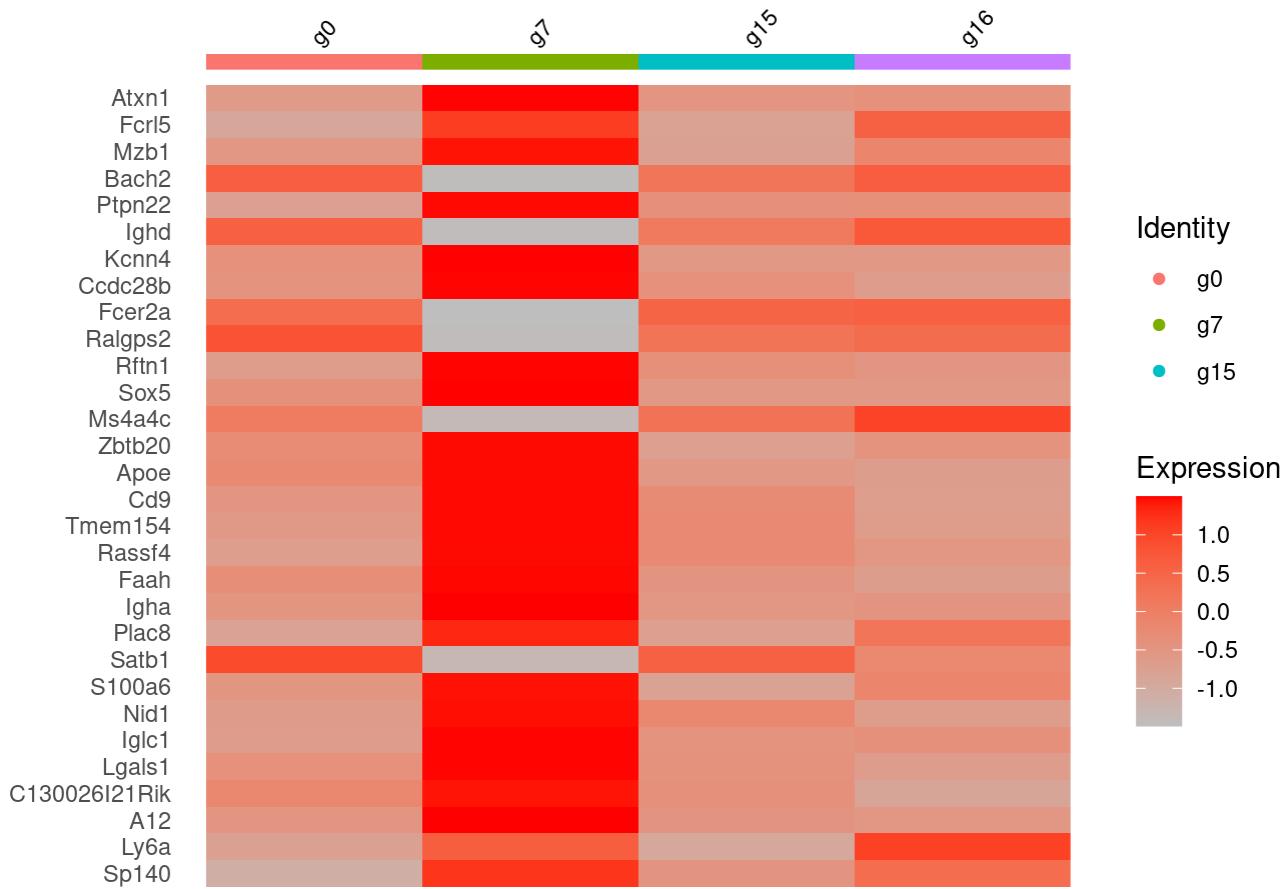
```
genes_of_interest <- row.names(average_expression@assays$RNA$scale.data)

expr_subset <- average_expression@assays$RNA$scale.data[genes_of_interest, ]

DoHeatmap(average_expression, features = row.names(five_top_markers_A12_uniq_pop_BM), size = 3
, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

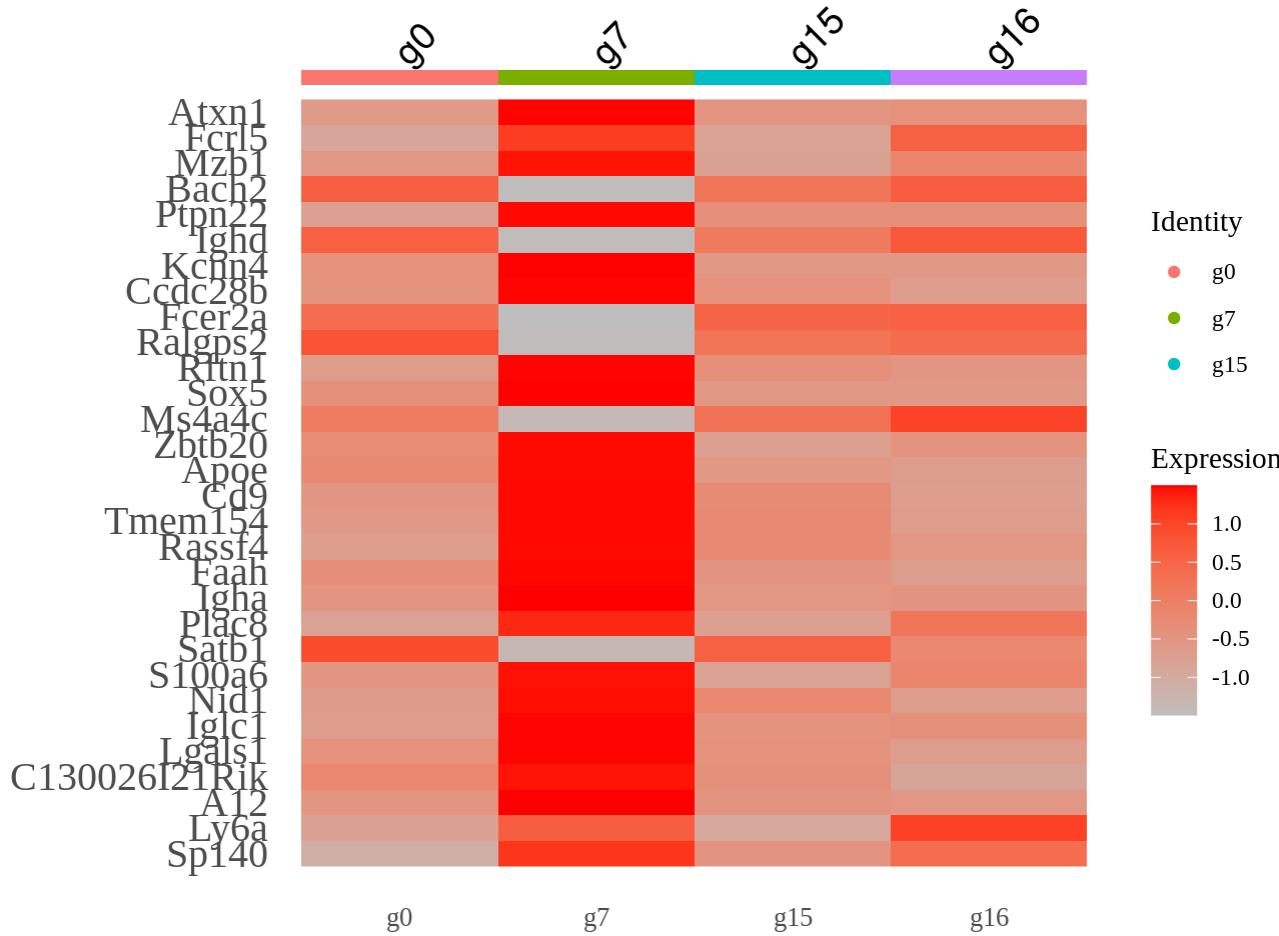


```
DoHeatmap(average_expression, features = row.names(five_top_markers_A12_uniq_pop_BM), group.colors = c(
  "ProB" = "#A52A2A",
  "Small PreB" = "lightblue3",
  "A12 unique" = "#E694C1",
  "Large PreB" = "#A694C1"
), size = 5, draw.lines = FALSE) +
scale_fill_gradientn(colors = c("grey", "red")) +
theme(
  text = element_text(family = "Times New Roman"),
  axis.text.x = element_text(size = 10, family = "Times New Roman"),
  axis.text.y = element_text(size = 15, family = "Times New Roman"),
  axis.title = element_text(size = 12, family = "Times New Roman"),
  plot.title = element_text(size = 14, face = "bold", family = "Times New Roman")
)
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 30 rows containing missing values or values outside the scale range
```

```
## (`geom_point()`).
```



```
average_expression <- AverageExpression(Seurat_Mature_A12, features = row.names(five_top_markers_A12_uniq_pop_BM), group.by = "seurat_clusters")$RNA %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  pivot_longer(cols = -gene, names_to = "cluster", values_to = "expression")
```

```
## Warning: The following 30 features were not found in the HTO assay: Atxn1,
## Fcrl5, Mzb1, Bach2, Ptpn22, Ighd, Kcnn4, Ccdc28b, Fcer2a, Ralgps2, Rftn1, Sox5,
## Ms4a4c, Zbtb20, Apoe, Cd9, Tmem154, Rassf4, Faah, Igaha, Plac8, Satb1, S100a6,
## Nid1, Igcl1, Lgals1, C130026I21Rik, A12, Ly6a, Sp140
## Warning: None of the features specified were found in the HTO assay.
```

```
five_top_markers_A12_uniq_pop_BM <- five_top_markers_A12_uniq_pop_BM %>%
  dplyr::mutate(gene = rownames(five_top_markers_A12_uniq_pop_BM))

data_for_dotplot_A12 <- data_for_dotplot %>%
  filter(cluster == "g7") %>%
  arrange(p_val_adj) %>%
  mutate(gene = factor(gene, levels = unique(gene)))

ggplot(data_for_dotplot_A12, aes(x = gene, y = cluster)) +
  geom_point(aes(size = significance, color = avg_log2FC)) +
```

```

scale_color_gradientn(
  colors = c("blue", "white", "red"),
  values = scales::rescale(c(-3, 0, 7)),
  name = "LogFC"
) +
scale_size_continuous(range = c(2, 8), name = "-log10(p-value)") +
labs(
  x = "Gene",
  y = "Cluster A12",
  title = "Dot plot of gene expression and significance for cluster A12",
  subtitle = "Size indicates -log10(p-value); Color indicates logFC"
) +
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 45, hjust = 1, family = "Times New Roman", size = 12),
  text = element_text(family = "Times New Roman"),
  legend.position = "right"
)

```

Dot plot of gene expression and significance for cluster A12

Size indicates -log10(p-value); Color indicates logFC

Cluster A12

Gene

```

Seurat_7_Mature <- subset(Seurat_Object_BM_selected_Bcells_idents, seurat_clusters %in% c("7"))
)
Idents(Seurat_7_Mature) <- Seurat_7_Mature$genotype
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_7_Mature, test.use = "roc", ident.1 =
= "A12", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)

```

```
## Warning: The following arguments are not used: thresh.use
```

We print the two most relevant per cluster:

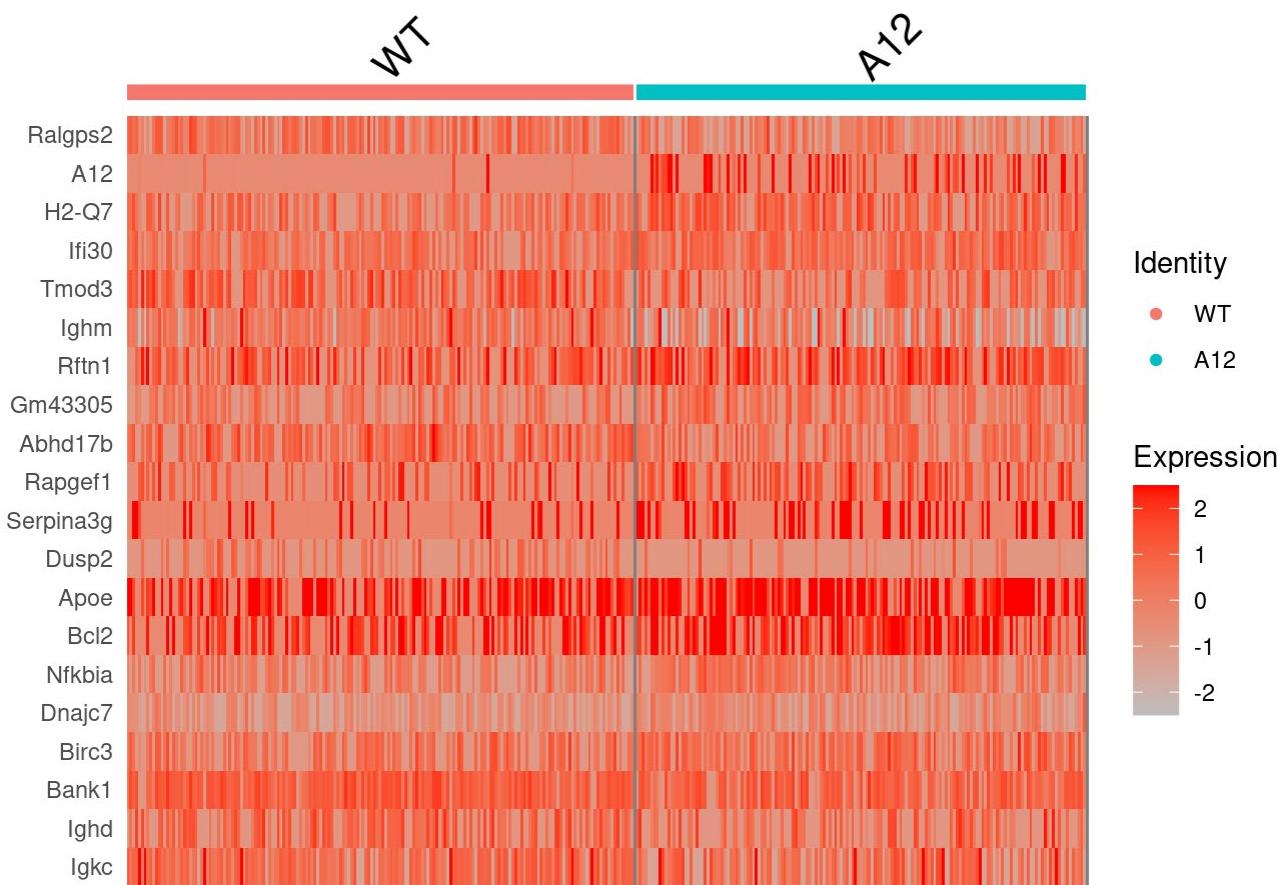
```
five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < -1) & (pct.1 > 0.25 | pct.
2 > 0.25)) %>%
  slice_max(n = 20, order_by = power) %>% # Seleccionar los 15 con mayo
r power
  slice_head(n = 20) %>%
  ungroup()

five_top_markers_A12_uniq_pop_BM
```

| | myAUC | avg_diff | power | avg_log2FC | pct.1 | pct.2 |
|--------------|-------|------------|-------|------------|-------|-------|
| ## Ralgps2 | 0.324 | -0.5265103 | 0.352 | -0.9390887 | 0.667 | 0.816 |
| ## A12 | 0.662 | 1.9201114 | 0.324 | 6.1734204 | 0.346 | 0.028 |
| ## H2-Q7 | 0.646 | 0.4847066 | 0.292 | 0.9243221 | 0.723 | 0.592 |
| ## Ifi30 | 0.628 | 0.3785994 | 0.256 | 0.6497149 | 0.836 | 0.704 |
| ## Tmod3 | 0.377 | -0.3888911 | 0.246 | -0.7852928 | 0.484 | 0.670 |
| ## Ighm | 0.387 | -0.2132819 | 0.226 | -0.3206135 | 0.799 | 0.966 |
| ## Rftn1 | 0.611 | 0.3125027 | 0.222 | 0.6010583 | 0.711 | 0.564 |
| ## Gm43305 | 0.609 | 0.3784453 | 0.218 | 0.7312291 | 0.654 | 0.475 |
| ## Abhd17b | 0.391 | -0.3409424 | 0.218 | -0.6702192 | 0.591 | 0.743 |
| ## Rapgef1 | 0.606 | 0.3060441 | 0.212 | 0.8047163 | 0.509 | 0.313 |
| ## Serpina3g | 0.604 | 0.4998727 | 0.208 | 1.6349043 | 0.346 | 0.151 |
| ## Dusp2 | 0.396 | -0.4320203 | 0.208 | -1.4579546 | 0.138 | 0.341 |
| ## Apoe | 0.603 | 0.4570317 | 0.206 | 0.7477424 | 0.742 | 0.615 |
| ## Bcl2 | 0.603 | 0.3226243 | 0.206 | 0.6609660 | 0.660 | 0.497 |
| ## Nfkbia | 0.602 | 0.2505510 | 0.204 | 0.5021966 | 0.704 | 0.547 |
| ## Dnajc7 | 0.601 | 0.4233637 | 0.202 | 0.7698870 | 0.780 | 0.654 |
| ## Birc3 | 0.600 | 0.2837484 | 0.200 | 0.5570012 | 0.748 | 0.587 |
| ## Bank1 | 0.400 | -0.2348335 | 0.200 | -0.3694460 | 0.836 | 0.950 |
| ## Ighd | 0.402 | -0.2743816 | 0.196 | -0.5092766 | 0.465 | 0.626 |
| ## Igkc | 0.402 | 0.1851711 | 0.196 | 0.2785540 | 0.717 | 0.894 |

```
DoHeatmap(Seurat_7_Mature, features =rownames(five_top_markers_A12_uniq_pop_BM)) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



```
average_expression <- AverageExpression(Seurat_7_Mature, features = row.names(five_top_markers_A12_uniq_pop_BM), return.seurat = TRUE)
```

```
## Warning: The following 20 features were not found in the HTO assay: Ralgps2,
## A12, H2-Q7, Ifi30, Tmod3, Igdm, Rftn1, Gm43305, Abhd17b, Rapgef1, Serpina3g,
## Dusp2, Apoe, Bcl2, Nfkbia, Dnajc7, Birc3, Bank1, Ighd, Igkc
```

```
## Warning: None of the features specified were found in the HTO assay.
```

```
## Centering and scaling data matrix
```

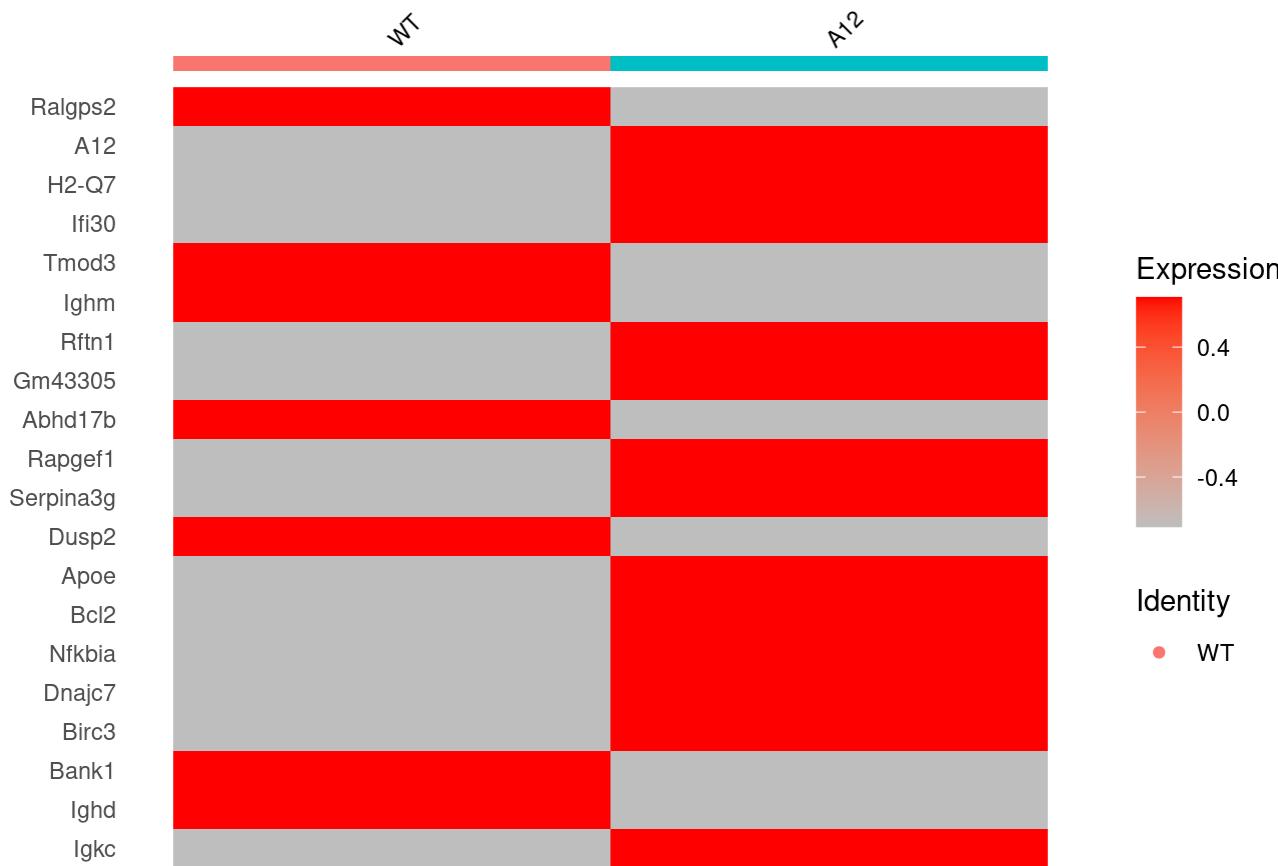
```
genes_of_interest <- row.names(average_expression@assays$RNA$scale.data)

expr_subset <- average_expression@assays$RNA$scale.data[genes_of_interest, ]

DoHeatmap(average_expression, features = row.names(five_top_markers_A12_uniq_pop_BM), size = 3
, draw.lines = FALSE) +
  scale_fill_gradientn(colors = c("grey", "red"))
```

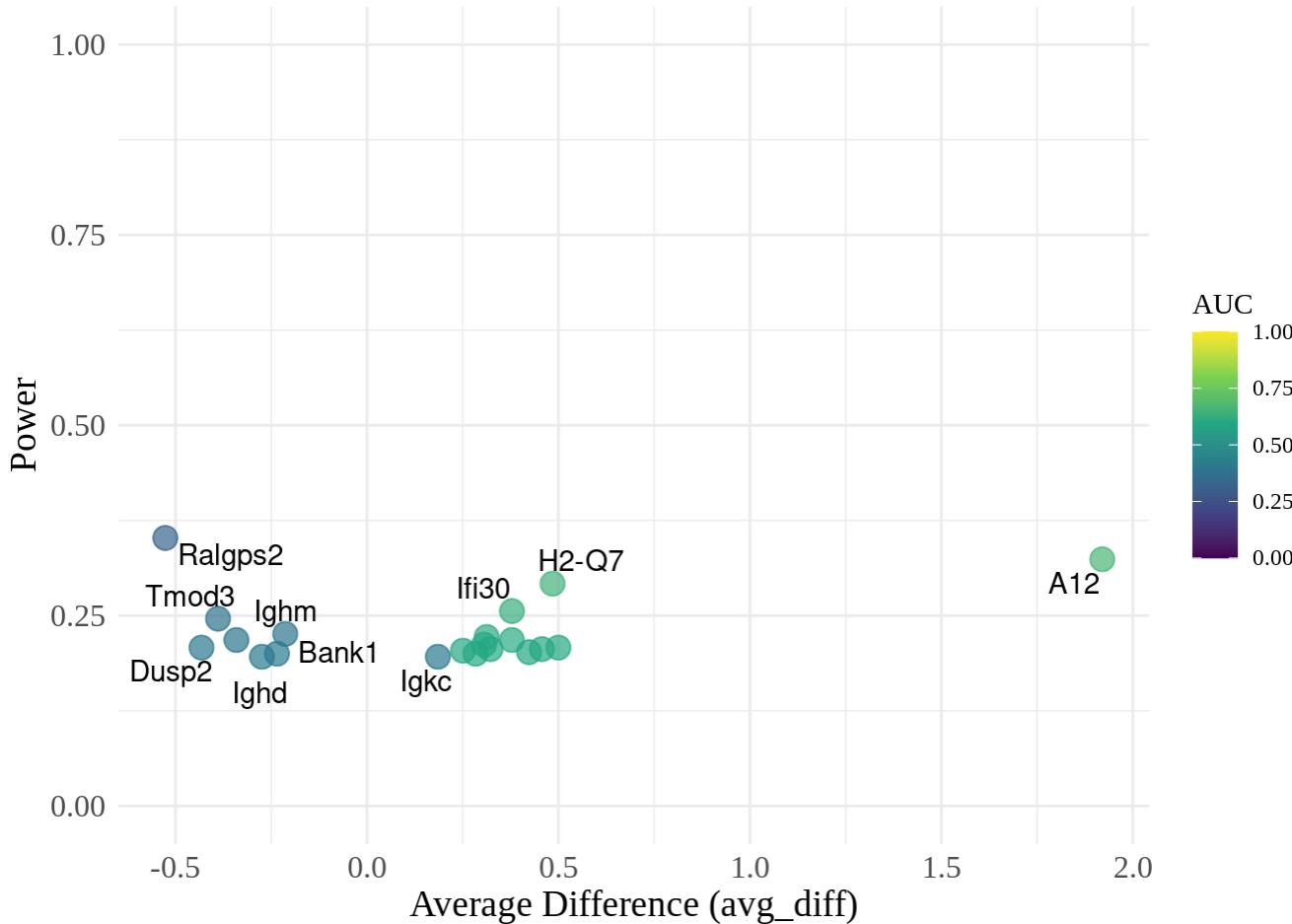
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
ggplot(five_top_markers_A12_uniq_pop_BM, aes(x = avg_diff, y = power, label = row.names(five_top_markers_A12_uniq_pop_BM))) +
  geom_point(aes(color = myAUC), alpha = 0.7, size = 4) + # Aumentar el tamaño de los puntos
  geom_text_repel() + # Etiquetas de genes
  scale_y_continuous(limits = c(0, 1), name = "Power") + # Escala para Power de 0 a 1
  scale_color_viridis_c(option = "D", name = "AUC", limits = c(0, 1)) + # Escala de color para AUC de 0 a 1
  labs(
    x = "Average Difference (avg_diff)"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14)
  )
```

```
## Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
all_markers_A12_uniq_pop_BM <- FindMarkers(object = Seurat_7_Mature, test.use = "MAST", ident.1 = "A12", min.pct = 0.25, min.diff.pct = 0.1, thresh.use = 0.25)
```

```
## Warning in new_with_repaired_slots(classname = method, design = colData(sca), : Dropping illegal slot(s) thresh.use for class BayesGLMlike.
## This likely indicates a bug in an upstream package.
```

```
##
## Done!
```

```
## Combining coefficients and standard errors
```

```
## Calculating log-fold changes
```

```
## Calculating likelihood ratio tests
```

```
## Refitting on reduced model...
```

```
##
## Done!
```

```

five_top_markers_A12_uniq_pop_BM <- all_markers_A12_uniq_pop_BM %>%
  dplyr::filter((avg_log2FC > 1 | avg_log2FC < -1) & (pct.1 > 0.25 | pct.
2 > 0.25) & p_val_adj < 1e-10) %>%
  slice_min(n = 30, order_by = p_val_adj) %>% # Seleccionar los 15 con
menor p_val_adj
  slice_head(n = 30) %>%
  ungroup()

five_top_markers_A12_uniq_pop_BM

```

```

##           p_val avg_log2FC pct.1 pct.2   p_val_adj
## A12 1.186252e-15    6.17342  0.346  0.028 2.358743e-11

```

```

DoHeatmap(Seurat_7_Mature, features =rownames(five_top_markers_A12_uniq_pop_BM), group.by = "seurat_clusters") +
  scale_fill_gradientn(colors = c("grey", "red"))

```

```

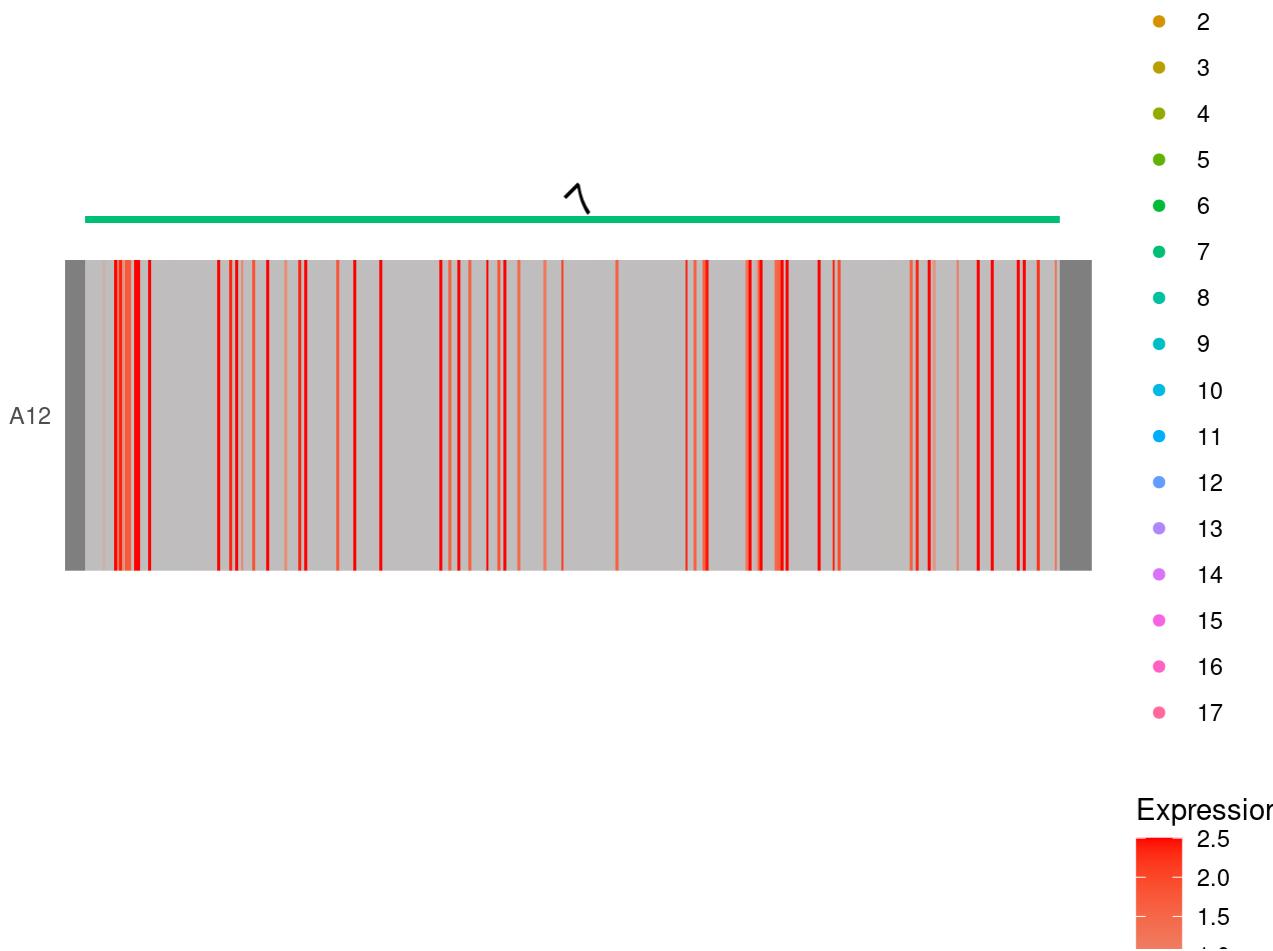
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

```

```

## Warning: Removed 17 rows containing missing values or values outside the scale range
## (`geom_text()`).

```



Proportions bone marrow per each cluster (numeric cluster designated by umap) grouped by cell subtype clusters

```
# Obtain percentages for each cluster in each mice
data_summary <- Seurat_Object_BM_selected_Bcells_idents@meta.data %>%
  group_by(genotype, mice, seurat_clusters) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
```

```
## `summarise()` has grouped output by 'genotype', 'mice'. You can override using
## the ` `.groups` argument.
```

```
# Store percentages in lists for each genotype

data_summary <- data_summary %>%
  filter(!is.na(seurat_clusters) & seurat_clusters != "17")
A12_test <- list()
WT_test <- list()
for (i in unique(data_summary$seurat_clusters)) {
  A12_test[[as.character(i)]] <- data_summary[data_summary$genotype == "A12" & data_summary$seurat_clusters == i, ]
  WT_test[[as.character(i)]] <- data_summary[data_summary$genotype == "WT" & data_summary$seurat_clusters == i, ]
}

# Crear una lista vacía para almacenar los resultados
resultados_t_test <- list()

# Iterar sobre las listas y realizar la prueba t para cada par de elementos
for (i in names(A12_test)) {
  grupo1 <- A12_test[[i]]$percentage
  grupo2 <- WT_test[[i]]$percentage

  # Realizar la prueba t
  resultado <- t.test(grupo1, grupo2, var.equal = TRUE)

  # Almacenar los resultados en la lista
  resultados_t_test[[paste("Comparacion", i)]] <- resultado
}

# Imprimir los resultados
names(resultados_t_test) <- sub("Comparacion ", "", names(resultados_t_test))

p_values <- sapply(resultados_t_test, function(res) res$p.value)
p_values
```

```
## 0.0255116908 0.0391106301 0.0370999799 0.5067537516 0.0005917738 0.0050339756
##       6        7        8        9        10       11
## 0.2996145404 0.9642276016 0.3769533720 0.0544918865 0.1465519753 0.0030589104
##      12       13       14       15       16
## 0.2192949155 0.0256244229 0.2097650032 0.0553496793 0.0575628903
```

```
# Calcular log fold change y añadir asteriscos de significancia
mean_per_cluster_A12 <- sapply(A12_test, function(df) mean(df$percentage, na.rm = TRUE))
mean_per_cluster_WT <- sapply(WT_test, function(df) mean(df$percentage, na.rm = TRUE))

p_value_l <- unlist(lapply(resultados_t_test, function(res) res$p.value))

# Calcular Log Fold Change (LFC)
log_fold_change <- log2(mean_per_cluster_A12 / mean_per_cluster_WT)

# Crear un data frame con todos los resultados
resultados_df <- data.frame(
  cluster = names(A12_test),
  mean_A12 = mean_per_cluster_A12,
  mean_WT = mean_per_cluster_WT,
  log_fold_change = log_fold_change,
  p_value = p_value_l
)

# Determinar niveles de significancia o incluir el valor de p si está entre 0.05 y 0.1
resultados_df$significance <- ifelse(
  resultados_df$p_value <= 0.001, "***",
  ifelse(
    resultados_df$p_value <= 0.01, "**",
    ifelse(
      resultados_df$p_value <= 0.05, "*",
      ifelse(
        resultados_df$p_value <= 0.1,
        sprintf("%.3f", resultados_df$p_value), # Formato para mostrar p-value con 3 decimales
        ""
      )
    )
  )
)

print(resultados_df)
```

| | cluster | mean_A12 | mean_WT | log_fold_change | p_value | significance |
|------|---------|------------|------------|-----------------|--------------|--------------|
| ## 0 | 0 | 13.3746409 | 33.4610369 | -1.32298200 | 0.0255116908 | * |
| ## 1 | 1 | 12.8048408 | 18.0422522 | -0.49469013 | 0.0391106301 | * |
| ## 2 | 2 | 17.1660740 | 12.6816888 | 0.43681324 | 0.0370999799 | * |
| ## 3 | 3 | 13.6520566 | 12.0547030 | 0.17952219 | 0.5067537516 | |
| ## 4 | 4 | 10.3119045 | 0.5367323 | 4.26396438 | 0.0005917738 | *** |
| ## 5 | 5 | 6.2896837 | 1.9278029 | 1.70602995 | 0.0050339756 | ** |
| ## 6 | 6 | 4.1731337 | 2.9156624 | 0.51730746 | 0.2996145404 | |
| ## 7 | 7 | 3.0024266 | 2.9685215 | 0.01638439 | 0.9642276016 | |
| ## 8 | 8 | 3.4562548 | 2.5196336 | 0.45599563 | 0.3769533720 | |

| | | | | | | |
|-------|----|-----------|-----------|-------------|--------------|-------|
| ## 9 | 9 | 3.3985266 | 1.4471341 | 1.23171083 | 0.0544918865 | 0.054 |
| ## 10 | 10 | 2.2850985 | 1.6426188 | 0.47625870 | 0.1465519753 | |
| ## 11 | 11 | 2.6063073 | 1.2538833 | 1.05560409 | 0.0030589104 | ** |
| ## 12 | 12 | 2.2471017 | 1.5631132 | 0.52364318 | 0.2192949155 | |
| ## 13 | 13 | 1.4352257 | 2.1124780 | -0.55765869 | 0.0256244229 | * |
| ## 14 | 14 | 1.9455073 | 1.4642412 | 0.40999319 | 0.2097650032 | |
| ## 15 | 15 | 0.9065086 | 1.4949271 | -0.72168251 | 0.0553496793 | 0.055 |
| ## 16 | 16 | 0.4348131 | 1.4481299 | -1.73572372 | 0.0575628903 | 0.058 |

```
# Agregar los niveles de significancia al resumen para usarlos en el gráfico
data_summary <- data_summary %>%
  left_join(resultados_df %>% select(cluster, significance), by = c("seurat_clusters" = "cluster"))

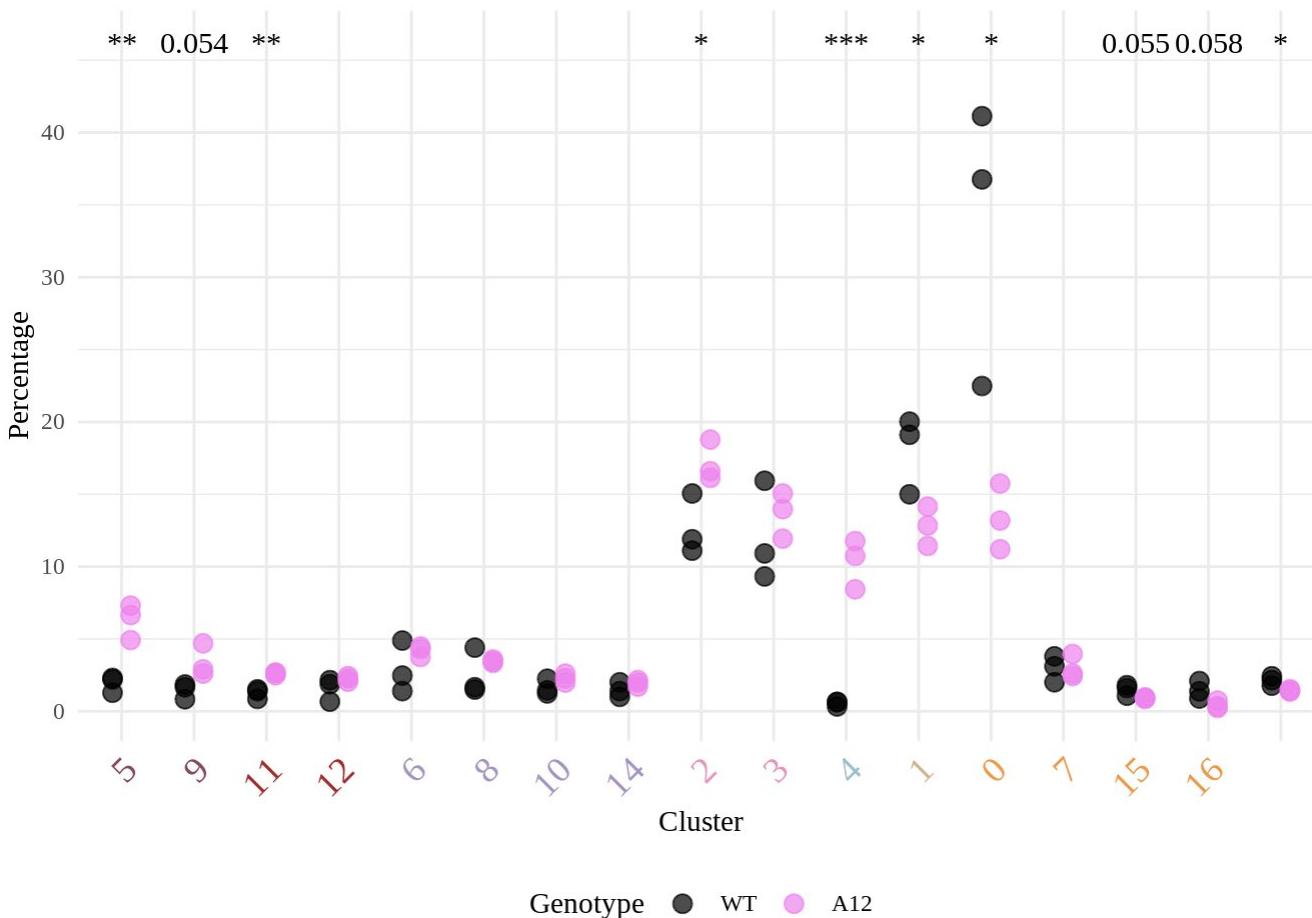
data_summary$seurat_clusters <- factor(data_summary$seurat_clusters)
# Paso 1: Asignar los grupos de clusters a data_summary
data_summary$cluster_group <- case_when(
  data_summary$seurat_clusters %in% c(5, 9) ~ "Pre-Pro B",
  data_summary$seurat_clusters %in% c(12, 11) ~ "ProB",
  data_summary$seurat_clusters %in% c(8, 6, 10, 14) ~ "Large PreB",
  data_summary$seurat_clusters %in% c(2, 3) ~ "Small PreB",
  data_summary$seurat_clusters == 4 ~ "A12 unique",
  data_summary$seurat_clusters == 1 ~ "Immature B",
  data_summary$seurat_clusters %in% c(0, 7, 15, 16) ~ "Mature B",
  TRUE ~ "Excluded" # Para el cluster 17 que se excluye
)

# Paso 2: Asignar colores a cada grupo con los colores específicos
cluster_colors <- c(
  "Pre-Pro B" = "palevioletred4",
  "ProB" = "#A52A2A",
  "Large PreB" = "#A694C1",
  "A12 unique" = "#E694C1",
  "Small PreB" = "lightblue3",
  "Immature B" = "tan",
  "Mature B" = "tan2"
)

# Paso 3: Convertir 'cluster_group' a un factor con el orden deseado
data_summary$cluster_group <- factor(data_summary$cluster_group,
  levels = c(
    "Pre-Pro B", "ProB", "Large PreB",
    "Small PreB", "A12 unique", "Immature B",
    "Mature B"
  )
)

# Convertir 'seurat_clusters' a factor para asegurar el orden de los clusters según el 'cluster_group'
data_summary$seurat_clusters <- factor(data_summary$seurat_clusters,
  levels = unique(data_summary$seurat_clusters[order(data_summary$cluster_group)]))
```

```
# Paso 4: Crear el gráfico
ggplot(data_summary, aes(x = seurat_clusters, y = percentage, color = genotype)) +
  geom_point(size = 3, position = position_dodge(width = 0.5), alpha = 0.7) +
  geom_text(
    data = resultados_df,
    aes(x = cluster, y = max(data_summary$percentage) + 5, label = significance),
    inherit.aes = FALSE,
    size = 4,
    family = "Times New Roman"
  ) +
  scale_color_manual(values = c("WT" = "black", "A12" = "violet")) # Colorear puntos por genotipo
  scale_x_discrete(
    breaks = levels(data_summary$seurat_clusters),
    labels = function(x) {
      # Asignar colores a las etiquetas del eje X según el grupo del cluster
      sapply(x, function(cluster) {
        group <- data_summary$cluster_group[data_summary$seurat_clusters == cluster][1]
        color <- cluster_colors[group]
        # Crear un texto con color y negrita (usando 'ggtext' para colores)
        return(paste0("<span style='color:", color, "; font-weight: bold;'>", cluster, "</span>"))
      })
    }
  ) +
  labs(
    x = "Cluster",
    y = "Percentage",
    color = "Genotype"
  ) +
  theme_minimal() +
  theme(
    text = element_text(family = "Times New Roman"),
    axis.text.x = element_markdown(size = 14, angle = 45, hjust = 1, family = "Times New Roman"),
    # Aumentamos el tamaño de la fuente y la negrita
    legend.position = "bottom"
  )
```



Anergy signature

```
# Genes obtained from anergic population RNaseq: # Multiple tolerance checkpoints restrain affinity maturation of B cells expressing the germlineprecursor of a lupus patient-derived anti-dsDNA antibody in knock-in mice
```

```
Anergic_marker_gene_list <- list(c("Fgl2", "Rg11", "Gcsam", "Slc9c1", "Serpina3f", "Snrnp25", "Lilrb4a",
  "Thsd7a", "Egr2", "Lrmp", "Serpina3g", "Marcks11", "Il9r", "Gm31763",
  "Socs1", "Cd72", "Serpina3g", "Ndrg1", "Lag3", "Ccdc17", "Anxa3",
  "Aox4", "Fut4", "Fbxl21", "Zbp1", "Apoe", "Ctse", "Wdfy1", "Cxcr4",
  "2210039B01Rik", "Gca", "Ptprv", "Chst7", "Tmem106b", "Fam3c", "Ephx1",
  "Dnm3", "Serpina6b", "Gins2", "Ptchd1", "H2-Oa", "Syt11", "Chst10",
  "Gbp7", "Fam129a", "H2-Q7", "Loxhd1", "Ackr3", "Traf1", "Ffar2", "Hivep3", "Rhbdfl",
  "Alcam",
  "Dusp4", "Btn2a2", "Mafb", "Camp", "Pard3b", "Nlrc5", "Slc2a6",
  "Chka", "2810417H13Rik", "Litaf", "Twsg1", "Lck", "Amigo2", "Tgfb3",
  "Hcrtr2", "Phf11a", "Glt28d2", "Jchain", "Ms4a6c", "Nsg1", "Iigp1",
  "Plin2", "Vav3", "Tmem151b", "Faah", "Snhg18", "Ccl4", "Cdca71",
  "Phf11b", "Vcam1", "Fam160a1", "Parm1", "Bend6", "Dtx3", "Batf",
  "Il12a", "Fscn1", "Mpo", "Dyrk4", "Hp", "Slco5a1", "Zfp30", "Daam1"))
```

```
Seurat_Object_SP_A12 <- AddModuleScore(
  object = Seurat_Object_SP_A12,
  features = Anergic_marker_gene_list,
```

```
  name = "Anergic_score"
)
```

```
## Warning: The following features are not present in the object: Slc9c1,
## 2210039B01Rik, Ptprv, 2810417H13Rik, Mpo, not searching for symbol synonyms
```

```
Seurat_Object_SP_WT <- AddModuleScore(
  object = Seurat_Object_SP_WT,
  features = Anergic_marker_gene_list,
  name = "Anergic_score"
)
```

```
## Warning: The following features are not present in the object: Slc9c1,
## 2210039B01Rik, Ptprv, 2810417H13Rik, Mpo, not searching for symbol synonyms
```

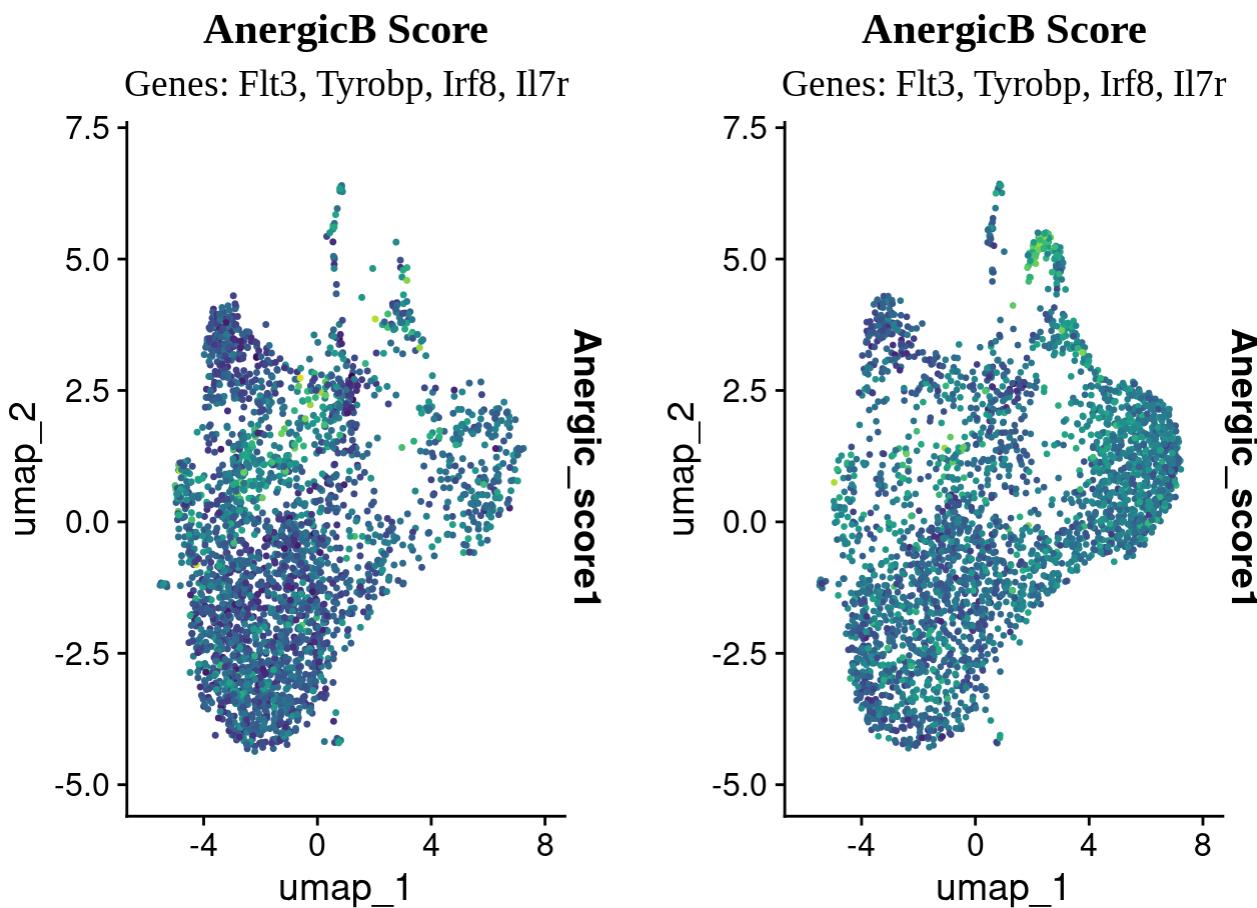
```
plot_A12 <- FeaturePlot(Seurat_Object_SP_WT, features = "Anergic_score1", split.by = "genotype"
") +
  scale_color_viridis(option = "D") +
  ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +
  theme(
    plot.title = element_text(family = "Times New Roman", size = 16),
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)
  )
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```
plot_WT <- FeaturePlot(Seurat_Object_SP_A12, features = "Anergic_score1", split.by = "genotype"
") +
  scale_color_viridis(option = "D") +
  ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +
  theme(
    plot.title = element_text(family = "Times New Roman", size = 16),
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)
  )
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```
wrap_plots(plot_A12, plot_WT, ncol = 2)
```



```
#Calculate mean anergic score
```

```
mean_scores_A12 <- Seurat_Object_SP_A12@meta.data %>%
  group_by(seurat_clusters_new_fin, mice) %>%
  summarise(mean_Anergic_score1 = mean(Anergic_score1, na.rm = TRUE))
```

`summarise()` has grouped output by 'seurat_clusters_new_fin'. You can override
using the `.groups` argument.

```
mean_scores_A12
```

```
## # A tibble: 21 × 3
## # Groups: seurat_clusters_new_fin [7]
##   seurat_clusters_new_fin mice mean_Anergic_score1
##   <fct>           <fct>     <dbl>
## 1 A12 expressing    A12_1     0.0781
## 2 A12 expressing    A12_2     0.0703
## 3 A12 expressing    A12_3     0.0753
## 4 Fo B             A12_1     -0.0304
## 5 Fo B             A12_2     -0.0265
## 6 Fo B             A12_3     -0.0305
## 7 MZ B             A12_1     -0.0118
## 8 MZ B             A12_2     -0.00831
## 9 MZ B             A12_3     -0.00652
```

```
## 10 Transitional B      A12_1      -0.0306
## # i 11 more rows
```

```
# Apply ANOVA for the variable Anergic_score1 for seurat_clusters_new_fin
anova_result_A12 <- aov(mean_Anergic_score1 ~ seurat_clusters_new_fin, data = mean_scores_A12)

# Obtain summary of ANOVA
anova_summary_A12 <- summary(anova_result_A12)

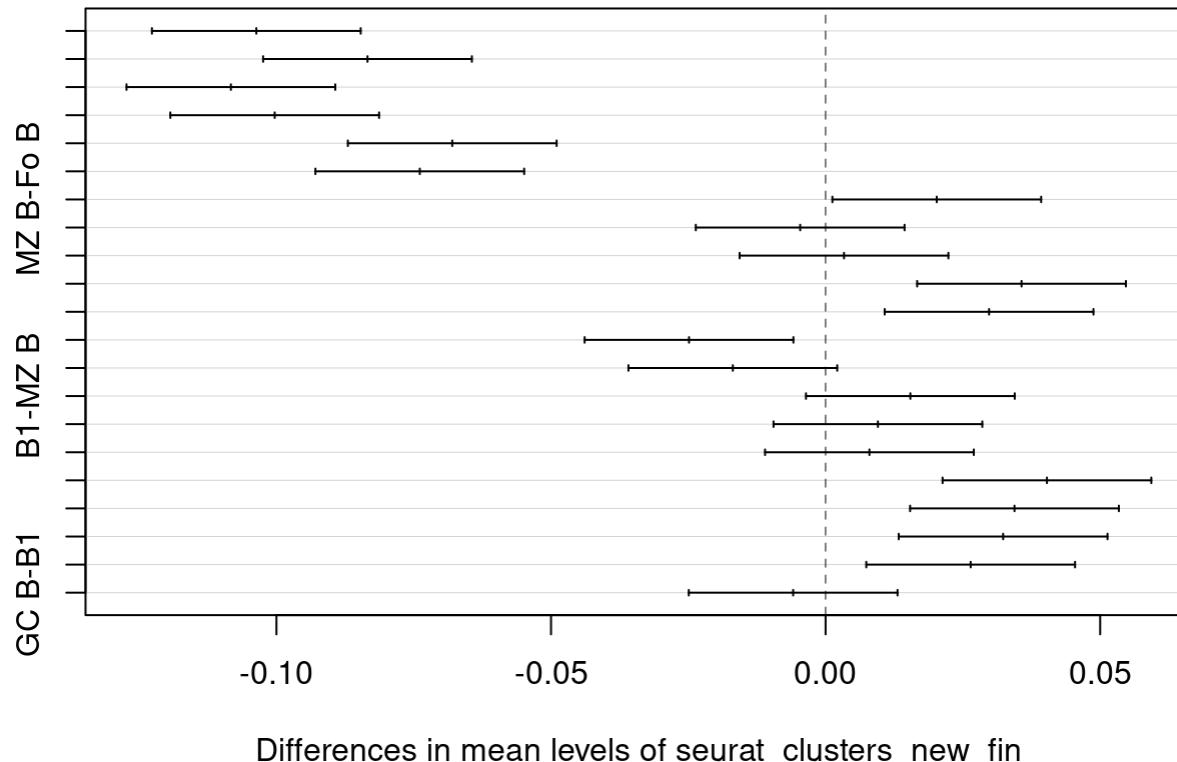
# Extract p values
anova_summary_A12[[1]]
```

```
##                                Df    Sum Sq   Mean Sq F value    Pr(>F)
## seurat_clusters_new_fin    6 0.0248841 0.0041473 89.227 2.402e-10 ***
## Residuals                  14 0.0006507 0.0000465
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Apply test of Tukey post-hoc
tukey_result_A12 <- TukeyHSD(anova_result_A12)

plot(tukey_result_A12)
```

95% family-wise confidence level



```
# Pairwise t-test
```

```
t_test_result_A12 <- pairwise.t.test(
  mean_scores_A12$mean_Anergic_score1,
  mean_scores_A12$seurat_clusters_new_fin,
  p.adjust.method = "BH"
)

t_test_pvals <- t_test_result_A12$p.value

significant_comparisons <- data.frame(
  group1 = rep(rownames(t_test_pvals), ncol(t_test_pvals)),
  group2 = rep(colnames(t_test_pvals), each = nrow(t_test_pvals)),
  p_value = as.vector(t_test_pvals)
)

significant_comparisons <- significant_comparisons %>%
  filter(p_value < 1e-6)

comparisons <- split(significant_comparisons, seq(nrow(significant_comparisons))) %>%
  lapply(function(x) c(x$group1, x$group2))

assign_asterisks <- function(p) {
  if (p < 0.001) {
    return("****")
  } else if (p < 0.01) {
    return("***")
  } else if (p < 0.05) {
    return("**")
  } else {
    return("ns") # No significativo
  }
}

annotations <- sapply(significant_comparisons$p_value, assign_asterisks)

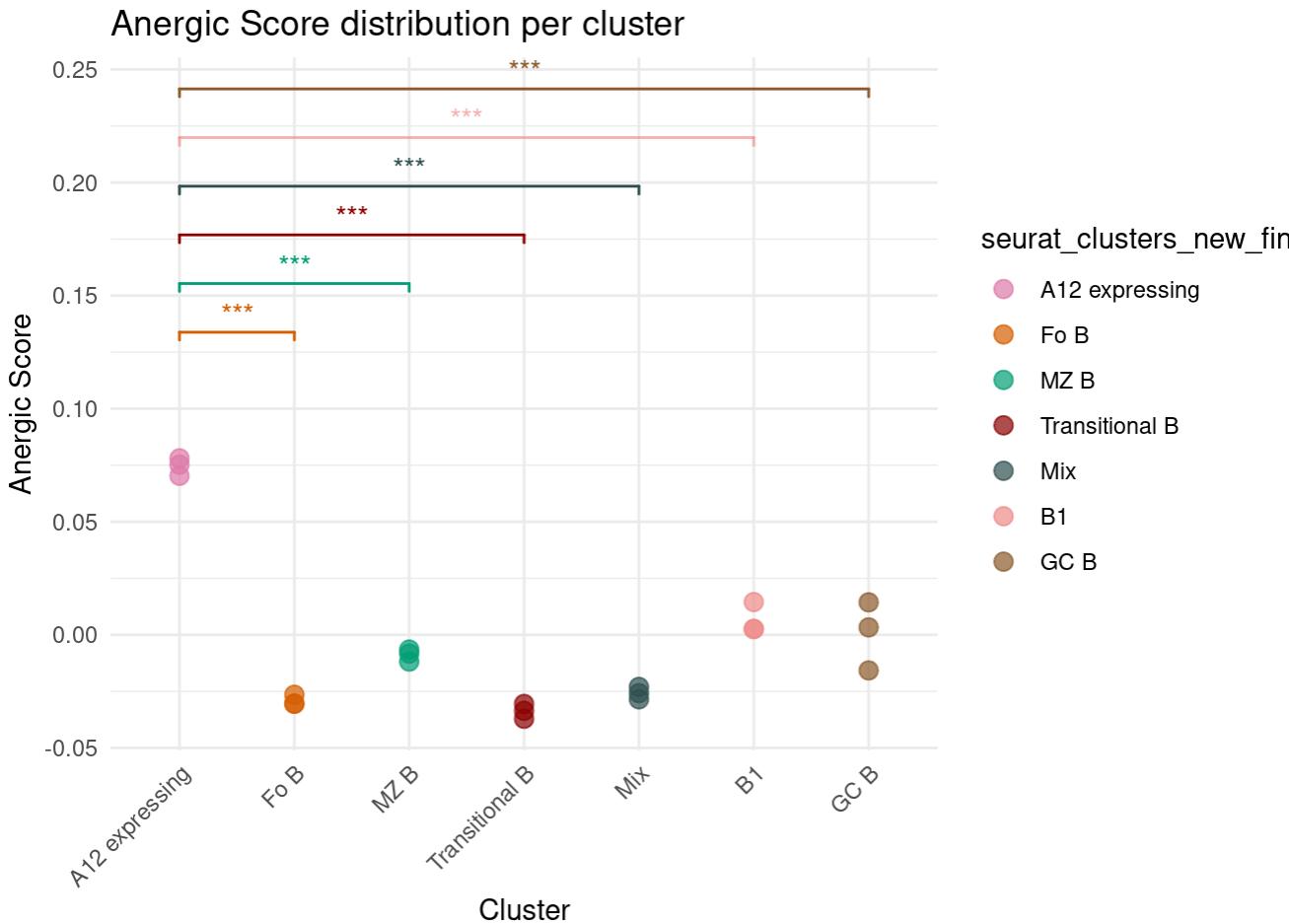
y_positions <- seq(from = max(mean_scores_A12$mean_Anergic_score1) + 0.05,
                     by = 0.01, length.out = length(comparisons))

cluster_colors <- c(
  "Fo B" = "#D55E00",
  "Transitional B" = "darkred",
  "MZ B" = "#009E73",
  "A12 expressing" = "#DC79A7",
  "Mix" = "darkslategray",
  "GC B" = "tan4",
  "B1" = "#ED8987A7"
)

ggplot(mean_scores_A12, aes(x = seurat_clusters_new_fin, y = mean_Anergic_score1, color = seurat_clusters_new_fin)) +
  geom_point(size = 3, alpha = 0.7) + # Puntos
  labs(x = "Cluster", y = "Anergic Score", title = "Anergic Score distribution per cluster") +
  theme_minimal() +
  scale_color_manual(values = cluster_colors) +

```

```
theme(axis.text.x = element_text(angle = 45, hjust = 1)) -  
geom_signif(  
  comparisons = comparisons,  
  annotations = annotations,  
  map_signif_level = FALSE,  
  y_position = y_positions,  
  step_increase = 0.1  
)
```



Anergic gene signature obtained from: IgD attenuates the IgM-induced anergy response in transitional and mature B cells

```

      "Alcam", "St8sia6", "Epha2", "D111", "Serp2", "D111", "Echdc3", "Astl", "Zfp948",
      "Syne1", "Echdc3", "Sh2d5", "8430427H17Rik", "Pacsin1", "Zfp52", "Sestd1", "Tym",
      "Zfp318", "Tacstd2", "Apon", "D111", "Specc1", "Zfp318", "Aqp3", "Csdc2", "D111",
      "Grap2", "Apon", "Rgs3", "Slc4a11", "Rassf6", "Fam167a", "B3gnt7", "Apon", "Apoe",
      "Renbp", "Apoe", "9630013D21Rik", "Apoe", "Apoe", "Dhrs3", "Apoe", "Sh2d5", "Klc3",
      "Apoe", "Apoe", "Apoe", "Abcb4", "Faim3", "Nab2", "Reep2", "Apoe", "Cldn10",
      "Cdk6", "Apoe", "Gm2447", "Klf10", "Atpl0a", "Usp2", "Renbp", "Mgll", "Il4il",
      "Nefh", "Omp", "Pdzd2", "Cacnald", "Marcks11", "Ccnd2", "X79554", "Dcbld1", "Dcbld1
",
      "Plxna3", "LOC100040974", "Vmnr2r60", "A_55_P2010134", "Ptchd1"))
)

```

```

Seurat_Object_SP_A12 <- AddModuleScore(
  object = Seurat_Object_SP_A12,
  features = Anergic_marker_gene_list,
  name = "Anergic_score"
)

```

```

## Warning: The following features are not present in the object: NAP114472-1,
## Lonrf3, LOC668306, NAP112057-1, Ppnrr, Lass6, A_55_P1961081, AY351698,
## 6330403A02Rik, 8430427H17Rik, Faim3, X79554, LOC100040974, Vmn2r60,
## A_55_P2010134, not searching for symbol synonyms

```

```

Seurat_Object_SP_WT <- AddModuleScore(
  object = Seurat_Object_SP_WT,
  features = Anergic_marker_gene_list,
  name = "Anergic_score"
)

```

```

## Warning: The following features are not present in the object: NAP114472-1,
## Lonrf3, LOC668306, NAP112057-1, Ppnrr, Lass6, A_55_P1961081, AY351698,
## 6330403A02Rik, 8430427H17Rik, Faim3, X79554, LOC100040974, Vmn2r60,
## A_55_P2010134, not searching for symbol synonyms

```

```

plot_A12 <- FeaturePlot(Seurat_Object_SP_WT, features = "Anergic_score1", split.by = "genotype
") +
  scale_color_viridis(option = "D") +
  ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +
  theme(
    plot.title = element_text(family = "Times New Roman", size = 16),
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)
  )

```

```

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.

```

```

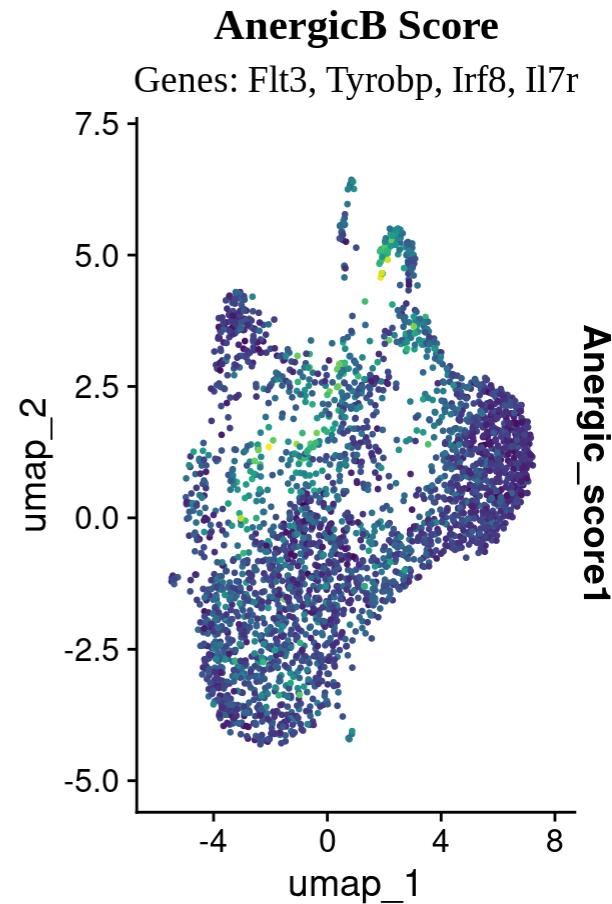
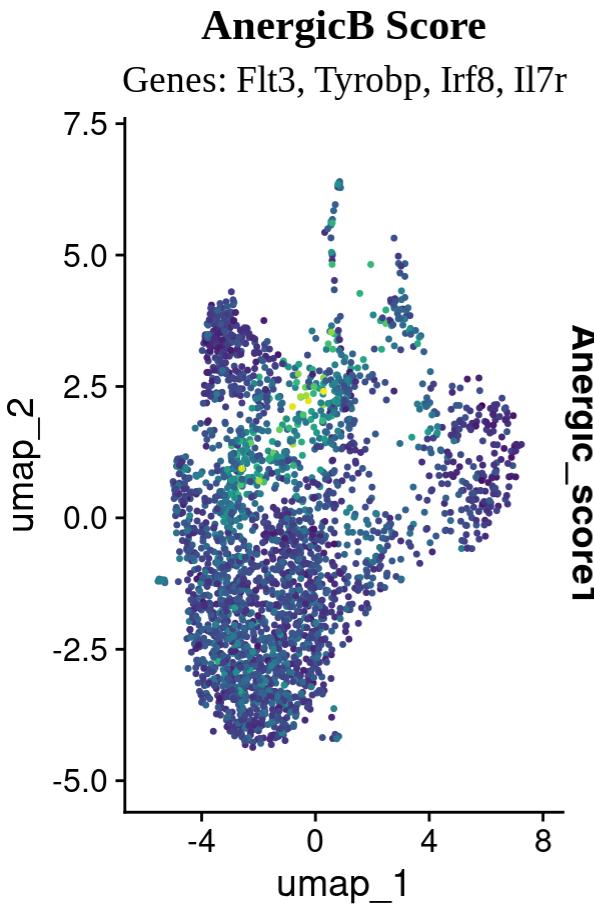
plot_WT <- FeaturePlot(Seurat_Object_SP_A12, features = "Anergic_score1", split.by = "genotype
") +
  scale_color_viridis(option = "D") +

```

```
ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +
  theme(
    plot.title = element_text(family = "Times New Roman", size = 16),
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)
  )
```

```
## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.
```

```
wrap_plots(plot_A12, plot_WT, ncol = 2)
```



```
mean_scores_A12 <- Seurat_Object_SP_A12@meta.data %>%
  group_by(seurat_clusters_new_fin, mice) %>%
  summarise(mean_Anergic_score1 = mean(Anergic_score1, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'seurat_clusters_new_fin'. You can override
## using the `.groups` argument.
```

```
mean_scores_A12
```

```
### A tibble: 21 × 3
### Groups: seurat_clusters_new_fin [7]
#>   seurat_clusters_new_fin mice mean_Anergic_score1
#>   <fct>                <dbl>             <dbl>
```

```

## <fct>      <fct>      <dbl>
## 1 A12 expressing    A12_1      0.0590
## 2 A12 expressing    A12_2      0.0574
## 3 A12 expressing    A12_3      0.0616
## 4 Fo B            A12_1     -0.0121
## 5 Fo B            A12_2     -0.0135
## 6 Fo B            A12_3     -0.0128
## 7 MZ B            A12_1     -0.0255
## 8 MZ B            A12_2     -0.0288
## 9 MZ B            A12_3     -0.0265
## 10 Transitional B   A12_1      0.00799
## # i 11 more rows

```

```

anova_result_A12 <- aov(mean_Anergic_score1 ~ seurat_clusters_new_fin, data = mean_scores_A12)

anova_summary_A12 <- summary(anova_result_A12)

anova_summary_A12[[1]]

```

```

##                               Df Sum Sq Mean Sq F value Pr(>F)
## seurat_clusters_new_fin  6 0.0140288 0.0023381 132.07 1.66e-11 ***
## Residuals                  14 0.0002479 0.0000177
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

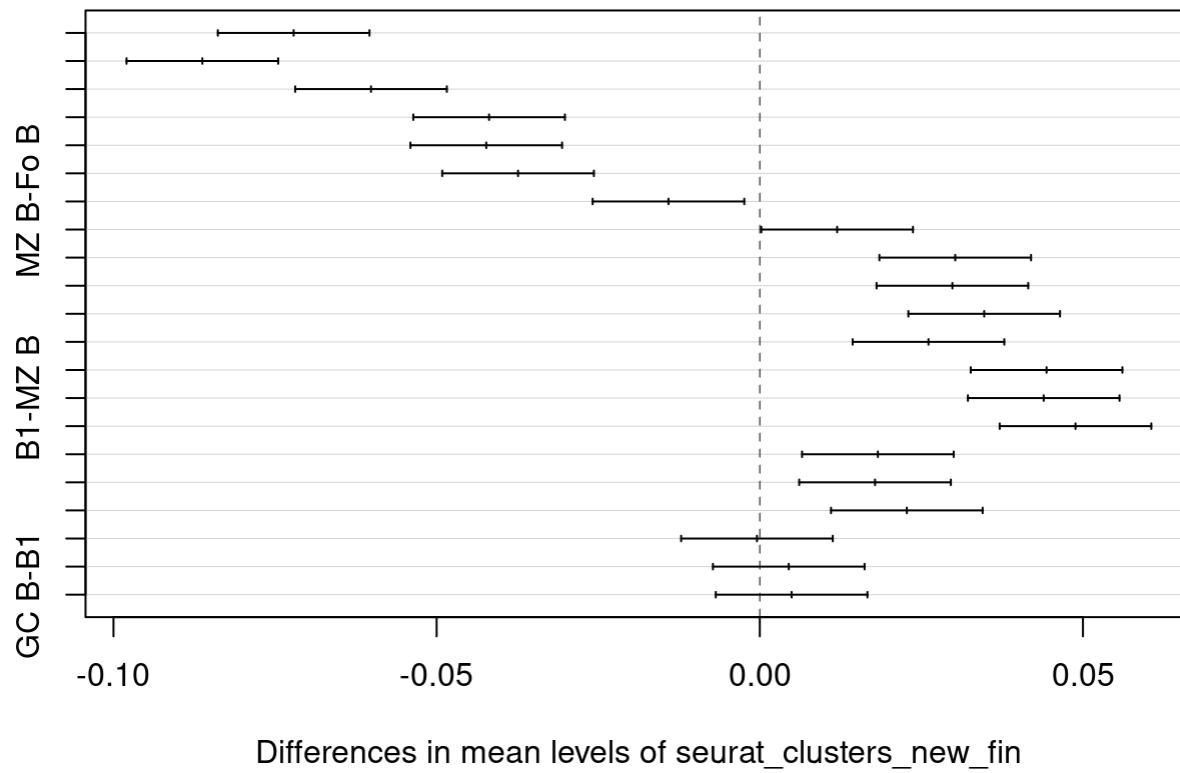
```

```

tukey_result_A12 <- TukeyHSD(anova_result_A12)
plot(tukey_result_A12)

```

95% family-wise confidence level



Differences in mean levels of seurat_clusters_new_fin

```
t_test_result_A12 <- pairwise.t.test(
  mean_scores_A12$mean_Anergic_score1,
  mean_scores_A12$seurat_clusters_new_fin,
  p.adjust.method = "BH"
)

t_test_pvals <- t_test_result_A12$p.value

significant_comparisons <- data.frame(
  group1 = rep(rownames(t_test_pvals), ncol(t_test_pvals)),
  group2 = rep(colnames(t_test_pvals), each = nrow(t_test_pvals)),
  p_value = as.vector(t_test_pvals)
)

significant_comparisons <- significant_comparisons %>%
  filter(p_value < 1e-6)

comparisons <- split(significant_comparisons, seq(nrow(significant_comparisons))) %>%
  lapply(function(x) c(x$group1, x$group2))

assign_asterisks <- function(p) {
  if (p < 0.001) {
    return("****")
  } else if (p < 0.01) {
    return("**")
  } else if (p < 0.05) {
    return("*")
  } else {
    return("n.s.")
  }
}
```

```

  return ("*")
} else {
  return ("ns") # No significativo
}
}

annotations <- sapply(significant_comparisons$p_value, assign_asterisks)

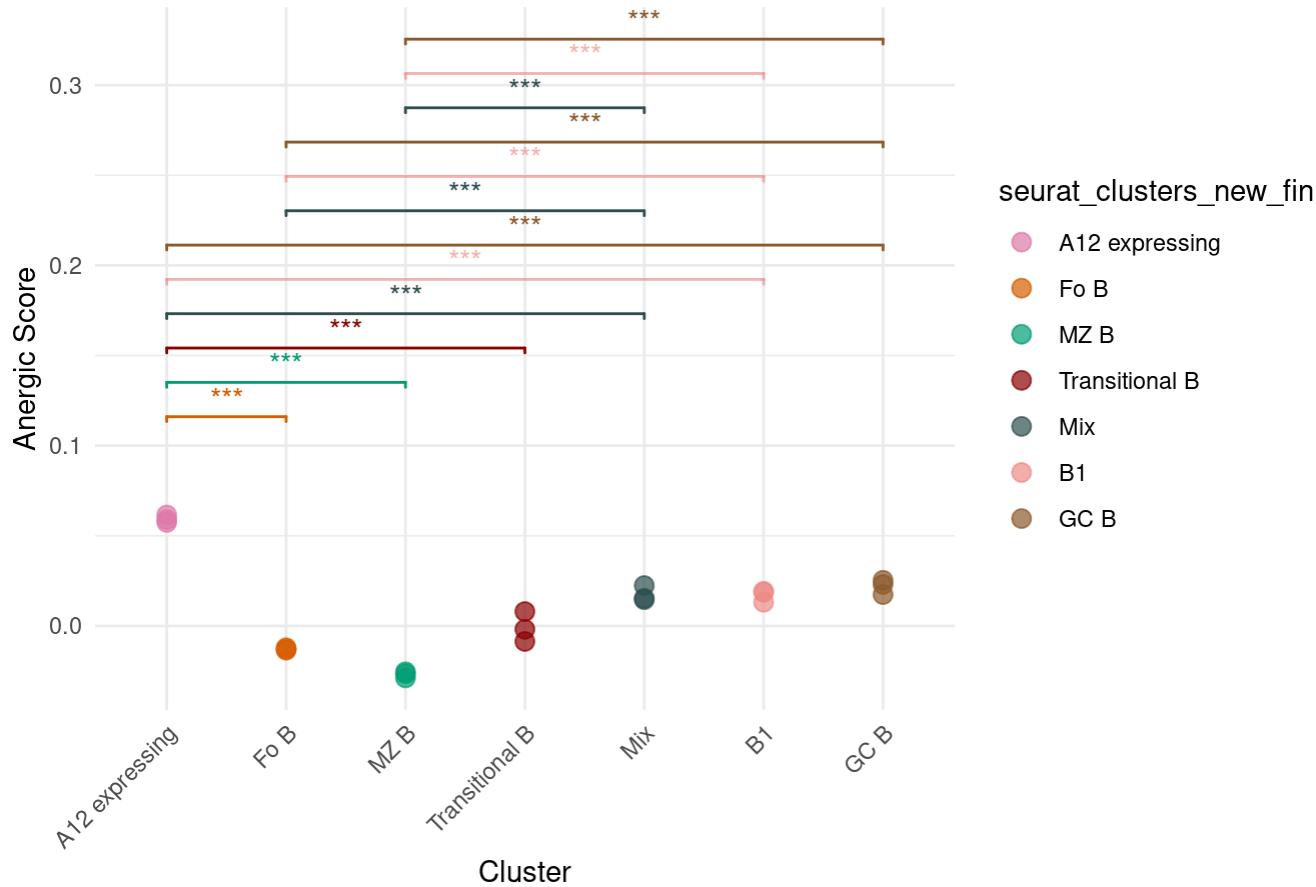
y_positions <- seq(from = max(mean_scores_A12$mean_Anergic_score1) + 0.05,
                     by = 0.01, length.out = length(comparisons))

cluster_colors <- c(
  "Fo B" = "#D55E00",
  "Transitional B" = "darkred",
  "MZ B" = "#009E73",
  "A12 expressing" = "#DC79A7",
  "Mix" = "darkslategray",
  "GC B" = "tan4",
  "B1" = "#ED8987A7"
)

ggplot(mean_scores_A12, aes(x = seurat_clusters_new_fin, y = mean_Anergic_score1, color = seurat_clusters_new_fin)) +
  geom_point(size = 3, alpha = 0.7) + # Puntos
  labs(x = "Cluster", y = "Anergic Score", title = "Distribución de Anergic Score por Clúster") +
  theme_minimal() +
  scale_color_manual(values = cluster_colors) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_signif(
    comparisons = comparisons,
    annotations = annotations,
    map_signif_level = FALSE,
    y_position = y_positions,
    step_increase = 0.1
)

```

Distribución de Anergic Score por Clúster



ABC marker gene list

```
ABC_marker_gene_list <- c("Itgb1", "Anxa2", "S100a10", "H2-Eb1", "Ly6c2", "Zbtb32", "Vim",
  "Sat1", "H2-Aa", "Ahnak", "Zbtb20", "Crip1", "Scimp", "Ly6a",
  "Cd74", "Tmsb4x", "Ass1", "Dnajc7", "Fcer1g", "Pdlim1",
  "Slc25a19", "Fxyd5", "Lgals3", "Fam3c", "Ctsh", "Rgs1", "Rcn3",
  "Tcf4", "Tagln2", "Ly6e", "Zeb2", "Laptm5", "Spn", "Sp140",
  "Lsp1", "Cd2")
```

```
Seurat_Object_SP_A12 <- AddModuleScore(
  object = Seurat_Object_SP_A12,
  features = Anergic_marker_gene_list,
  name = "Anergic_score"
)
```

```
## Warning: The following features are not present in the object: NAP114472-1,
## Lonrf3, LOC668306, NAP112057-1, Ppnrr, Lass6, A_55_P1961081, AY351698,
## 6330403A02Rik, 8430427H17Rik, Faim3, X79554, LOC100040974, Vmn2r60,
## A_55_P2010134, not searching for symbol synonyms
```

```
Seurat_Object_SP_WT <- AddModuleScore(
  object = Seurat_Object_SP_WT,
  features = Anergic_marker_gene_list,
```

```
name = "Anergic_score"  
)
```

```
## Warning: The following features are not present in the object: NAP114472-1,  
## Lonrf3, LOC668306, NAP112057-1, Ppnr, Lass6, A_55_P1961081, AY351698,  
## 6330403A02Rik, 8430427H17Rik, Faim3, X79554, LOC100040974, Vmn2r60,  
## A_55_P2010134, not searching for symbol synonyms
```

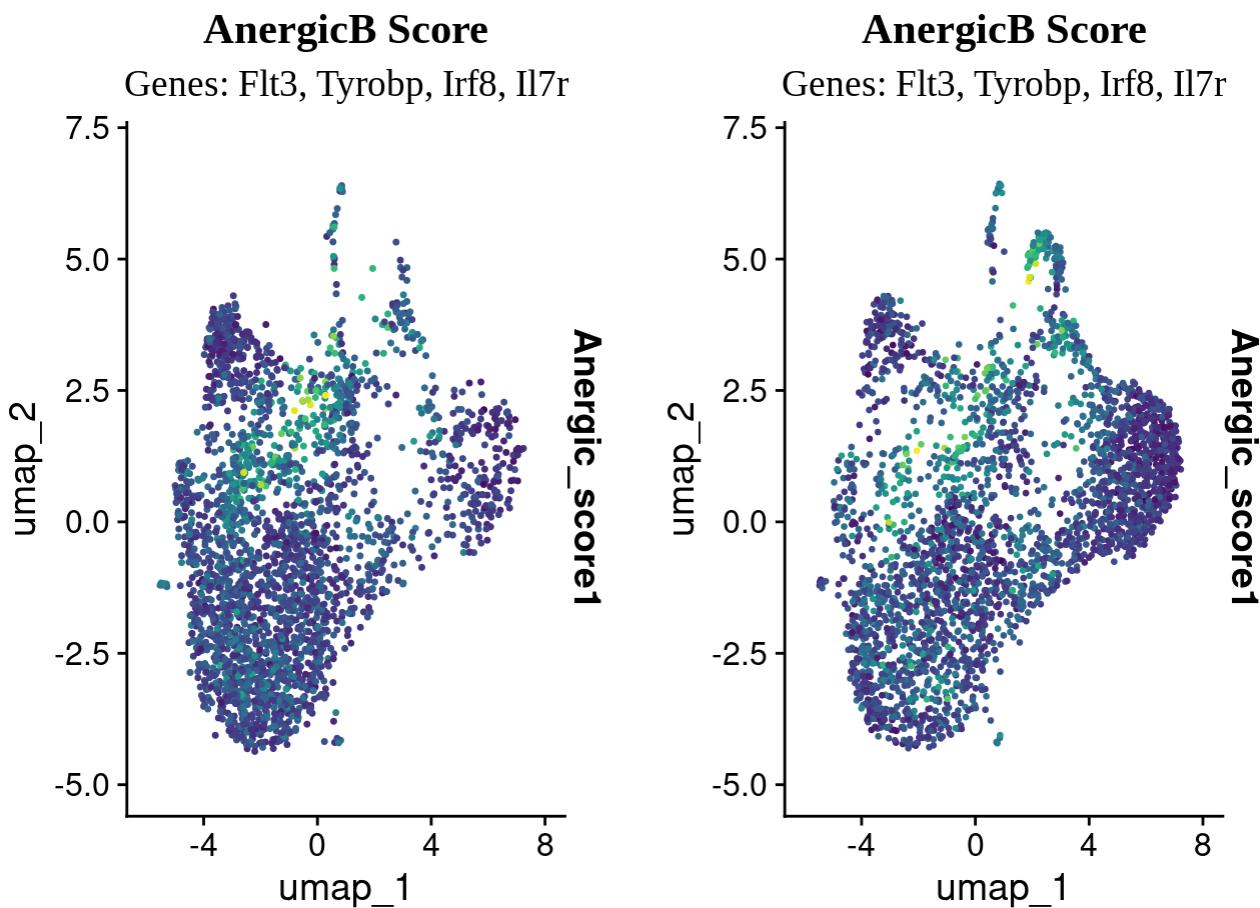
```
plot_A12 <- FeaturePlot(Seurat_Object_SP_WT, features = "Anergic_score1", split.by = "genotype") +  
  scale_color_viridis(option = "D") +  
  ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +  
  theme(  
    plot.title = element_text(family = "Times New Roman", size = 16),  
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)  
)
```

```
## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.
```

```
plot_WT <- FeaturePlot(Seurat_Object_SP_A12, features = "Anergic_score1", split.by = "genotype") +  
  scale_color_viridis(option = "D") +  
  ggtitle("AnergicB Score", subtitle = "Genes: Flt3, Tyrobp, Irf8, Il7r") +  
  theme(  
    plot.title = element_text(family = "Times New Roman", size = 16),  
    plot.subtitle = element_text(family = "Times New Roman", size = 14, hjust = 0.5)  
)
```

```
## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.
```

```
wrap_plots(plot_A12, plot_WT, ncol = 2)
```



```
mean_scores_A12 <- Seurat_Object_SP_A12@meta.data %>%
  group_by(seurat_clusters_new_fin, mice) %>%
  summarise(mean_Anergic_score1 = mean(Anergic_score1, na.rm = TRUE))
```

`summarise()` has grouped output by 'seurat_clusters_new_fin'. You can override
using the `.groups` argument.

```
mean_scores_A12
```

```
## # A tibble: 21 × 3
## # Groups: seurat_clusters_new_fin [7]
##   seurat_clusters_new_fin mice mean_Anergic_score1
##   <fct>           <fct>     <dbl>
## 1 A12 expressing    A12_1     0.0590
## 2 A12 expressing    A12_2     0.0574
## 3 A12 expressing    A12_3     0.0616
## 4 Fo B             A12_1    -0.0121
## 5 Fo B             A12_2    -0.0135
## 6 Fo B             A12_3    -0.0128
## 7 MZ B             A12_1    -0.0255
## 8 MZ B             A12_2    -0.0288
## 9 MZ B             A12_3    -0.0265
## 10 Transitional B A12_1     0.00799
## # ... with 11 more rows
```

```
anova_result_A12 <- aov(mean_Anergic_score1 ~ seurat_clusters_new_fin, data = mean_scores_A12)
anova_summary_A12 <- summary(anova_result_A12)

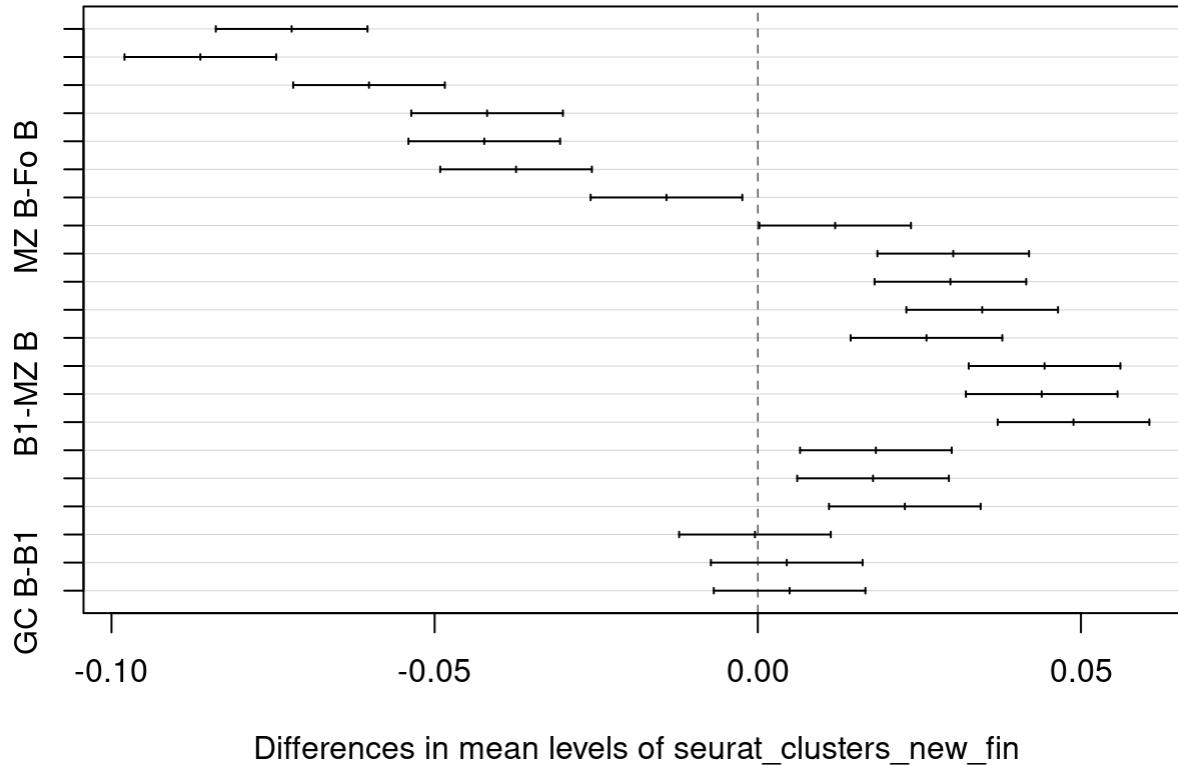
anova_summary_A12[[1]]
```

```
##                               Df      Sum Sq   Mean Sq F value    Pr(>F)
## seurat_clusters_new_fin   6 0.0140288 0.0023381 132.07 1.66e-11 ***
## Residuals                  14 0.0002479 0.0000177
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
tukey_result_A12 <- TukeyHSD(anova_result_A12)

plot(tukey_result_A12)
```

95% family-wise confidence level



```
t_test_result_A12 <- pairwise.t.test(
  mean_scores_A12$mean_Anergic_score1,
  mean_scores_A12$seurat_clusters_new_fin,
  p.adjust.method = "BH"
)
```

```
t_test_pvals <- t_test_result_A12$p.value

significant_comparisons <- data.frame(
  group1 = rep(rownames(t_test_pvals), ncol(t_test_pvals)),
  group2 = rep(colnames(t_test_pvals), each = nrow(t_test_pvals)),
  p_value = as.vector(t_test_pvals)
)

significant_comparisons <- significant_comparisons %>%
  filter(p_value < 0.05)

comparisons <- split(significant_comparisons, seq(nrow(significant_comparisons))) %>%
  lapply(function(x) c(x$group1, x$group2))

assign_asterisks <- function(p) {
  if (p < 0.001) {
    return("****")
  } else if (p < 0.01) {
    return("***")
  } else if (p < 0.05) {
    return("**")
  } else {
    return("*ns")
  }
}

annotations <- sapply(significant_comparisons$p_value, assign_asterisks)
y_positions <- seq(from = max(mean_scores_A12$mean_Anergic_score1) + 0.05,
                     by = 0.01, length.out = length(comparisons))
cluster_colors <- c(
  "Fo B" = "#D55E00",
  "Transitional B" = "darkred",
  "MZ B" = "#009E73",
  "A12 expressing" = "#DC79A7",
  "Mix" = "darkslategray",
  "GC B" = "tan4",
  "B1" = "#ED8987A7"
)

ggplot(mean_scores_A12, aes(x = seurat_clusters_new_fin, y = mean_Anergic_score1, color = seurat_clusters_new_fin)) +
  geom_point(size = 3, alpha = 0.7) + # Puntos
  labs(x = "Cluster", y = "Anergic Score", title = "Anergic score distribution per cluster") +
  theme_minimal() +
  scale_color_manual(values = cluster_colors) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_signif(
    comparisons = comparisons,
    annotations = annotations,
    map_signif_level = FALSE,
    y_position = y_positions,
    step_increase = 0.1
  )

```

Anergic score distribution per cluster

