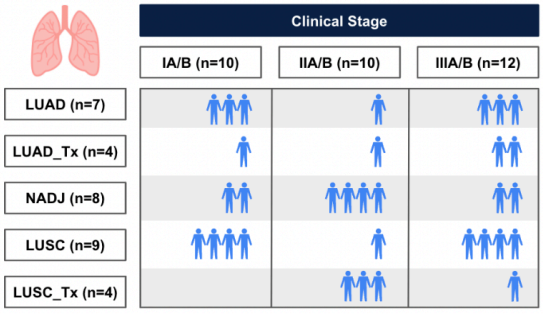
Analysis of scRNA-seq Data Comparing Non Small Cell Lung Cancer to Normal

Here I analyze single cell RNA-sequencing (scRNA-seq) data from 32 samples over 5 biological conditions: lung adenocarcinoma (treated and untreated), lung squamous cell carcinoma (treated and untreated), and normal adjacent cells.

32 flash frozen tissues from two major subtypes of treated and untreated Non Small Cell Lung Cancer (NSCLC) (Lung Adenocarcinoma and Lung Squamous Cell Carcinoma) and Normal Adjacent samples were obtained by 10x Genomics from BioIVT ASTERAND.

Sample size



Probe Barcode per sample

Probe Barcode	Samples 1-16	Samples 17-32
BC001	LUSC_IIIA	LUSC_Tx_IIB
BC002	LUAD_IIB	LUAD_Tx_IA2
BC003	LUSC_IIIB	LUSC_IA3
BC004	LUSC_IIIB	LUSC_Tx_IIB
BC005	LUAD_IIIA	LUSC_IB
BC006	NADJ_IA3	LUAD_IA3
BC007	NADJ_IIIA	LUSC_IIB
BC008	LUAD_IA2	LUSC_IA3
BC009	LUAD_IIIA	LUAD_Tx_IIIA
BC010	NADJ_IIB	LUAD_IB
BC011	NADJ_IIB	LUSC_IA2
BC012	NADJ_IIA	LUSC_IIIA
BC013	LUAD_Tx_IIA	NADJ_IB
BC014	LUAD_Tx_IIIA	NADJ_IIIA
BC015	LUSC_Tx_IIIA	LUAD_IIIA
BC016	NADJ_IIB	LUSC_Tx_IIB

(Left) Sample sizes by tumor subtype and clinical stage. The NSCLC tumor subtypes are displayed by row: Lung Adenocarcinoma (LUAD); Treated Lung Adenocarcinoma (LUAD_Tx); Normal Adjacent (NADJ); Lung Squamous Cell Carcinoma (LUSC); and Treated Lung Squamous Cell Carcinoma (LUSC_Tx). Clinical stages are displayed by column.

(Right) 32 samples were split into two unique 16-plex pools and labeled as Samples 1-16 and Samples 17-32. The two right hand columns list the samples within each 16-plex pool, notated by tumor subtype and clinical stage. The left column lists the Probe Barcode associated with a particular sample.

Tumor tissues were fixed for 20 hours at 4°C following the demonstrated protocol Tissue Fixation & Dissociation for Chromium Fixed RNA Profiling (CG000553).

The Fixed RNA Gene Expression library was generated as described in the Chromium Fixed RNA Profiling for Multiplexed Samples User Guide (CG000527). Samples were run as a 16-sample multiplexed experiment, in which each sample was hybridized with a unique Probe Barcode in a separate hybridization reaction. After hybridization, samples were pooled in equal proportions, washed, and then run in replicates across four GEM lanes each. The resulting libraries were sequenced on an Illumina NovaSeq6000 with approximately 20k read pairs per cell.

- Total 897,733 cells detected
    - NADJ: 254,429
    - LUAD: 168,978
    - LUAD_Tx: 123,642
    - LUSC: 225,536
    - LUSC_Tx: 125,148
- Paired-end, dual indexing: 28 cycles Read 1, 10 cycles i7, 10 cycles i5, 90 cycles Read 2

The sample-level web summaries from the `cellranger multi` runs are available for download here.

Note: This dataset has Low Post-Normalization Read Depth due to a large discrepancy in the sequencing depth per cell between samples in the library.

In [57]:
```
# load packages
library(dplyr)
library(Seurat)
library(SeuratObject)
library(SeuratDisk)
library(BPCells)
library(Azimuth)
library(future)

library(DESeq2)

library(pheatmap)
library(RColorBrewer)
library(ggplot2)
library(EnhancedVolcano)

# change the current plan to access parallelization
```

```
plan("multisession", workers = availableCores())
plan()

# increase size of plots from default
options(repr.plot.width = 14,
        repr.plot.height = 14) # from 7, 7
```

```
Loading required package: ggrepel
```

```
structure(function (..., workers = c(system = 12), envir = parent.fram
e())
strategy(..., workers = workers, envir = envir), class = c("FutureStra
tegy",
"tweaked", "multisession", "cluster", "multiprocess", "future",
"function"), init = "done", untweakable = "persistent", call = plan("m
ultisession",
    workers = availableCores()))
```

In [6]:
```
# load data from h5 file
file_path <- "data/16plex_900k_32_NSCLC_multiplex_count_filtered_feature_bc_
nsclc_data <- open_matrix_10x_hdf5(
    path = file_path
)

# Write the matrix to a directory
mat_dir <- "data/nsclc_counts"
write_matrix_dir(
    mat = nsclc_data,
    dir = mat_dir)

# Now that we have the matrix on disk, we can load it
nsclc_mat <- open_matrix_dir(dir = mat_dir)
nsclc_mat <- Azimuth:::ConvertEnsembleToSymbol(mat = nsclc_mat, species = "h

# Create Seurat Object
nsclc <- CreateSeuratObject(counts = nsclc_mat)
```

```
18082 x 897733 IterableMatrix object with class MatrixDir

Row names: ENSG00000187634, ENSG00000188976 ... ENSG00000198727
Col names: AAACAAGCAAACGGTCACTTTAGG-1, AAACAAGCAATAAGGAACTTTAGG-1 ... TTTGTG
AGTGGATGGTATTCGGTT-128

Data type: uint32_t
Storage order: column major

Queued Operations:
1. Load compressed matrix from directory /home/fmbuga/scrna/projects/nsclc_1
0x_900k/data/nsclc_counts
```

```
Attaching package: 'dplyr'


The following object is masked from 'package:biomaRt':

    select


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

    filter, lag


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

    intersect, setdiff, setequal, union



Batch submitting query [=======>----------------------]  25% eta: 10s

Batch submitting query [===============>--------------]  50% eta:  8s

Batch submitting query [======================>-------]  75% eta:  4s
```

In [2]:
```
# Now that we have the matrix on disk, we can load it
mat_dir <- "data/nsclc_counts"
nsclc_mat <- open_matrix_dir(dir = mat_dir)
nsclc_mat <- Azimuth:::ConvertEnsembleToSymbol(mat = nsclc_mat, species = "h

# Create Seurat Object
(nsclc <- CreateSeuratObject(counts = nsclc_mat))
```

```
An object of class Seurat
18037 features across 897733 samples within 1 assay
Active assay: RNA (18037 features, 0 variable features)
 1 layer present: counts
```

add sample metadata

In [4]:
```
nsclc[['sample']] <- as.integer(sub('.*-', '', colnames(nsclc)))
sample_table <- read.csv('data/16plex_900k_32_NSCLC_multiplex_aggregation.cs
sample_table$sample <- (1:nrow(sample_table))
nsclc[[]] <- merge(nsclc[[]], sample_table, by = "sample", all.x = TRUE)
nsclc[[]]
```

	orig.ident	nCount_RNA	nFeature_RNA	sample
	<fct>	<dbl>	<dbl>	<int>
AAACAAGCAAACGGTCACTTTAGG-1	SeuratProject	14792	5177	1
AAACAAGCAATAAGGAACTTTAGG-1	SeuratProject	2067	1464	1
AAACAAGCAGCACTAAACTTTAGG-1	SeuratProject	4371	2515	1
AAACAAGCATCGATAGACTTTAGG-1	SeuratProject	8413	3827	1
AAACCAATCCGAAAGTACTTTAGG-1	SeuratProject	2460	1232	1
AAACCAATCCTCAAGCACTTTAGG-1	SeuratProject	1261	953	1
AAACCAATCGATATAGACTTTAGG-1	SeuratProject	547	469	1
AAACCAGGTATTGGGAACTTTAGG-1	SeuratProject	1802	971	1
AAACCAGGTGACATTCACTTTAGG-1	SeuratProject	2949	1740	1
AAACCAGGTTACTTCTACTTTAGG-1	SeuratProject	810	682	1
AAACCAGGTTGGATGAACTTTAGG-1	SeuratProject	15537	4091	1
AAACCAGGTTTAGTTGACTTTAGG-1	SeuratProject	495	346	1
AAACCGGTCAGGCAATACTTTAGG-				

1	SeuratProject	2059	1159	1
AAACCGGTCATGATGAACTTTAGG-1	SeuratProject	720	596	1
AAACCGGTCATGCATTACTTTAGG-1	SeuratProject	442	338	1
AAACCGGTCGCTCAACACTTTAGG-1	SeuratProject	685	496	1
AAACCGGTCGCTTCGGACTTTAGG-1	SeuratProject	2651	1411	1
AAACGGGCAACCTTGTACTTTAGG-1	SeuratProject	1153	887	1
AAACGGGCACACCCACACTTTAGG-1	SeuratProject	766	546	1
AAACGTTCAAGCCTCCACTTTAGG-1	SeuratProject	3353	1510	1
AAACGTTCAAGTCGCGACTTTAGG-1	SeuratProject	855	699	1
AAACGTTCAATCGTTCACTTTAGG-1	SeuratProject	3587	1997	1
AAACGTTCAATCTCTAACTTTAGG-1	SeuratProject	708	604	1
AAACGTTCATGGCCGAACTTTAGG-1	SeuratProject	1394	954	1
AAACTGGGTCACCAGCACTTTAGG-1	SeuratProject	2913	1674	1
AAACTGGGTCGTCCAAACTTTAGG-1	SeuratProject	917	753	1
AAACTGGGTGGCTTTGACTTTAGG-1	SeuratProject	3625	1438	1
AAACTGGGTTTACTGAACTTTAGG-1	SeuratProject	445	399	1
AAACTGTCACTAAGCAACTTTAGG-1	SeuratProject	392	348	1
AAAGATGCACACTGGGACTTTAGG-1	SeuratProject	1357	1024	1
⋮	⋮	⋮	⋮	⋮
TTTCATGCAGACTCAAATTCGGTT-128	SeuratProject	12677	5160	128 1

TTTCATGCATTTGTTGATTCGGTT-128	SeuratProject	1001	710	128	1
TTTCGCGCATATGGTGATTCGGTT-128	SeuratProject	1020	714	128	1
TTTGAGAAGCGAATCGATTCGGTT-128	SeuratProject	2803	1579	128	1
TTTGAGAAGCTTACCAATTCGGTT-128	SeuratProject	662	518	128	1
TTTGAGAAGGCGCAGTATTCGGTT-128	SeuratProject	431	347	128	1
TTTGAGAAGGCTTAGGATTCGGTT-128	SeuratProject	407	364	128	1
TTTGCCCGTAGGTGACATTCGGTT-128	SeuratProject	8196	4104	128	1
TTTGCCCGTGAACGGTATTCGGTT-128	SeuratProject	1604	1098	128	1
TTTGCCCGTGCGATAAATTCGGTT-128	SeuratProject	321	155	128	1
TTTGCCCGTGTAATGGATTCGGTT-128	SeuratProject	6934	3807	128	1
TTTGCCCGTTAGTTTGATTCGGTT-128	SeuratProject	1923	842	128	1
TTTGCGCAGACATAGCATTCGGTT-128	SeuratProject	436	389	128	1
TTTGCGCAGCAAATTGATTCGGTT-128	SeuratProject	395	209	128	1
TTTGCGCAGCAACATAATTCGGTT-128	SeuratProject	1922	1089	128	1
TTTGCGCAGTCGAAGTATTCGGTT-128	SeuratProject	329	299	128	1
TTTGCGCAGTCGTTCAATTCGGTT-128	SeuratProject	2097	1558	128	1
TTTGCGCAGTTATGGAATTCGGTT-128	SeuratProject	1927	1361	128	1
TTTGCGGGTGACACAAATTCGGTT-128	SeuratProject	21377	6601	128	1
TTTGCTCTCAACCGGAATTCGGTT-128	SeuratProject	5979	3107	128	1

TTTGCTCTCCTGGTTAATTCGGTT-128	SeuratProject	310	282	128	1
TTTGCTCTCGCAAATAATTCGGTT-128	SeuratProject	597	345	128	1
TTTGCTCTCGGGACTAATTCGGTT-128	SeuratProject	897	698	128	1
TTTGCTGAGCTAAACAATTCGGTT-128	SeuratProject	1325	940	128	1
TTTGCTGAGCTAGCCTATTCGGTT-128	SeuratProject	6749	3354	128	1
TTTGCTGAGGTCAAGTATTCGGTT-128	SeuratProject	251	235	128	1
TTTGCTGAGTTTCCAGATTCGGTT-128	SeuratProject	2830	1666	128	1
TTTGGACGTAAGCTAGATTCGGTT-128	SeuratProject	9822	4359	128	1
TTTGGACGTTCGAATCATTCGGTT-128	SeuratProject	309	272	128	1
TTTGTGAGTGGATGGTATTCGGTT-128	SeuratProject	2053	1339	128	1

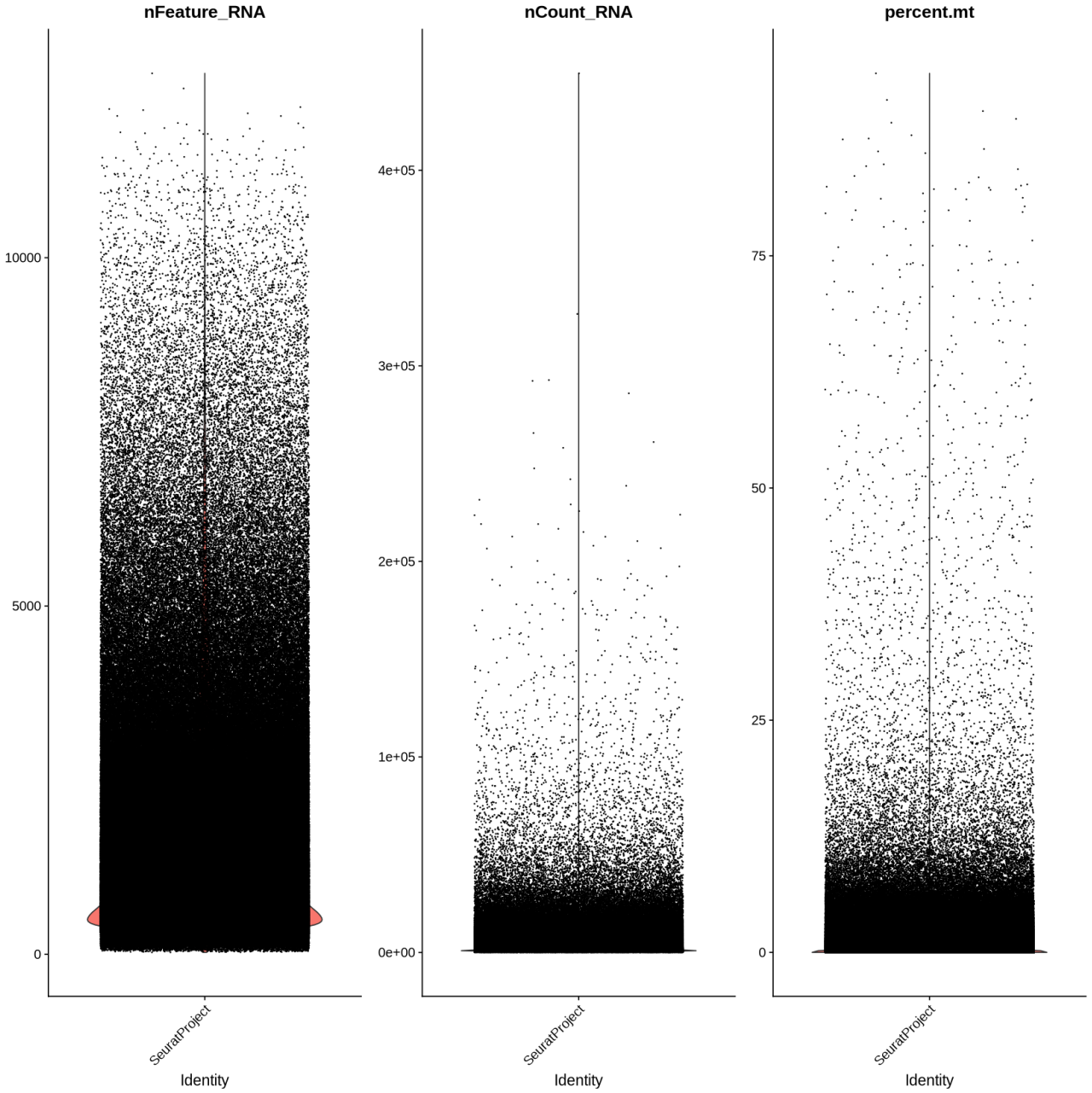
QC

In [5]:
```
# mitochondrial counts to metadata
nsclc[["percent.mt"]] <- PercentageFeatureSet(nsclc, pattern = "^MT-")
summary(nsclc[["percent.mt"]])
```

```
  percent.mt
Min.   : 0.0000
1st Qu.: 0.1459
Median : 0.5726
Mean   : 1.1015
3rd Qu.: 1.3354
Max.   :94.6472
```

In [6]:
```
# Visualize QC metrics as a violin plot
VlnPlot(nsclc,
        features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
        ncol = 3)
```

```
Warning message:
"Default search for "data" layer in "RNA" assay yielded no results; utilizin
g "counts" layer instead."
```



In [7]:
```
summary(nsclc[[c("nFeature_RNA", "nCount_RNA", "percent.mt")]])
```

```
 nFeature_RNA     nCount_RNA       percent.mt      
Min.   :   28   Min.   :    45   Min.   : 0.0000  
1st Qu.:  626   1st Qu.:   800   1st Qu.: 0.1459  
Median : 1051   Median :  1510   Median : 0.5726  
Mean   : 1450   Mean   :  3072   Mean   : 1.1015  
3rd Qu.: 1761   3rd Qu.:  2987   3rd Qu.: 1.3354  
Max.   :12646   Max.   :449595   Max.   :94.6472  
```

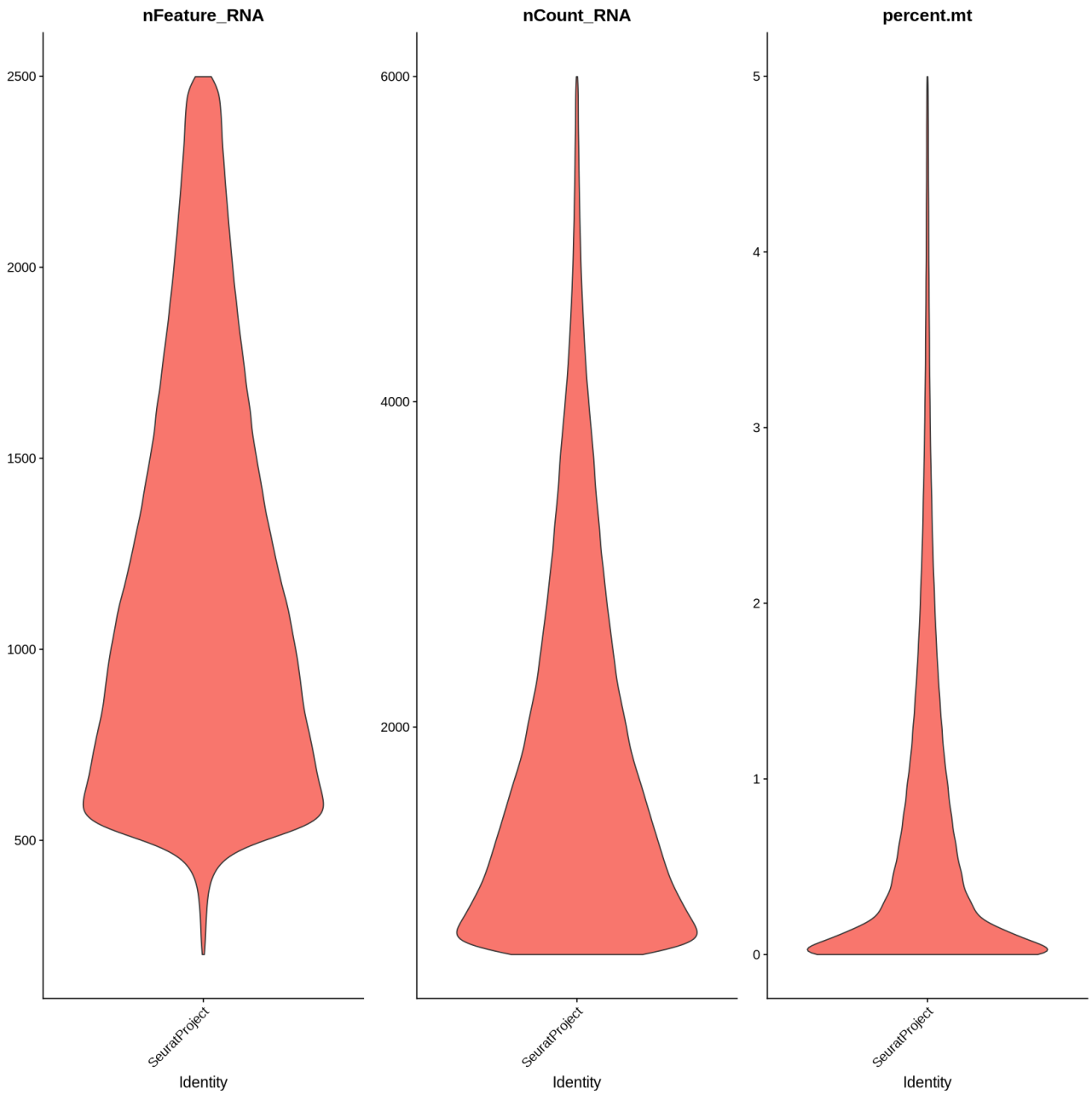
In [8]:
```
# filter out cells with too low/high counts, too high mitochondrial counts,
nsclc
nsclc <- subset(nsclc,
                subset = nFeature_RNA > 200 & nFeature_RNA < 2500 &
                           percent.mt < 5 &
                           nCount_RNA > 600 & nCount_RNA < 6000)
nsclc
summary(nsclc[[c("nFeature_RNA", "nCount_RNA", "percent.mt")]])
```

```
An object of class Seurat
18037 features across 897733 samples within 1 assay
Active assay: RNA (18037 features, 0 variable features)
 1 layer present: counts
An object of class Seurat
18037 features across 619727 samples within 1 assay
Active assay: RNA (18037 features, 0 variable features)
 1 layer present: counts
  nFeature_RNA    nCount_RNA     percent.mt
 Min.   : 201   Min.   : 601   Min.   :0.00000
 1st Qu.: 749   1st Qu.: 987   1st Qu.:0.09719
 Median :1063   Median :1532   Median :0.43732
 Mean   :1170   Mean   :1832   Mean   :0.78538
 3rd Qu.:1507   3rd Qu.:2421   3rd Qu.:1.13821
 Max.   :2499   Max.   :5999   Max.   :4.99802
```

The filtering step leaves us with about 620,000 cells.

In [9]:
```
# visualize QC metrics post-filter
VlnPlot(nsclc,
        features = c("nFeature_RNA", "nCount_RNA", "percent.mt"),
        alpha = 0,
        ncol = 3)
```

```
Warning message:
"Default search for "data" layer in "RNA" assay yielded no results; utilizin
g "counts" layer instead."
```


nFeature_RNA
nCount_RNA
percent.mt
2500
2000
1500
1000
500
6000
4000
2000
5
4
3
2
1
0
SeuratProject
SeuratProject
SeuratProject
Identity
Identity
Identity

data normalization, selection of highly variable genes, data scaling, dimension reduction, clustering, and visualization

Next, we normalize the data, select highly variable features, and scale the data using the `SCTransform` method.

We then cluster the data and visualize using UMAP.

This is submitted as a SLURM job to the HPC.

In [3]:
```
# SLURM submission script
file_to_read <- "~/tools/slurm_scripts/Rscript_submit_himem_1node_12cpus_mem
file_content <- readLines(file_to_read)
cat(file_content, sep = "\n")
```

```
#!/bin/bash
#
#SBATCH --job-name=Rscript_submit_himem_1node_12cpus_mem500_231125
#SBATCH --output=/home/fmbuga/tools/slurm_scripts/slurm_out_err/Rscript_subm
it_himem_1node_12cpus_mem500_231125_%j.out
#SBATCH --error=/home/fmbuga/tools/slurm_scripts/slurm_out_err/Rscript_submi
t_himem_1node_12cpus_mem500_231125_%j.err
#
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=12
#SBATCH --mem=500G
#SBATCH --partition=himem


datenow=$(date)
echo $datenow
srun hostname

start=$(date +%s)
echo "start time: $start"
echo $HOSTNAME


eval "$(conda shell.bash hook)"
conda activate seurat5
echo ""
conda info
echo ""
conda list
echo ""
```

```
echo ""
echo "Path to R used: "
which R
R --version


###

killall R

Rscript \
    --save \
    --verbose \
    $1 \
    $2 \
    $3 \
    $4

###

echo ""
srun hostname


echo ""
end=$(date +%s)
echo "end time: $end"
runtime_s=$(echo $(( end - start )))
echo "total run time(s): $runtime_s"
sec_per_min=60
sec_per_hr=3600
runtime_m=$(echo "scale=2; $runtime_s / $sec_per_min;" | bc)
echo "total run time(m): $runtime_m"
runtime_h=$(echo "scale=2; $runtime_s / $sec_per_hr;" | bc)
echo "total run time(h): $runtime_h"
```

In [2]:
```
# R script for data normalization, selection of highly variable genes, data
file_to_read <- "~/tools/R_scripts/scrnaseq_sctransform_to_umap_231127.r"
file_content <- readLines(file_to_read)
cat(file_content, sep = "\n")
```

```
# clear workspace
rm(list = ls())
gc()

args <- commandArgs(TRUE) # to access command line arguments
print(args)
```

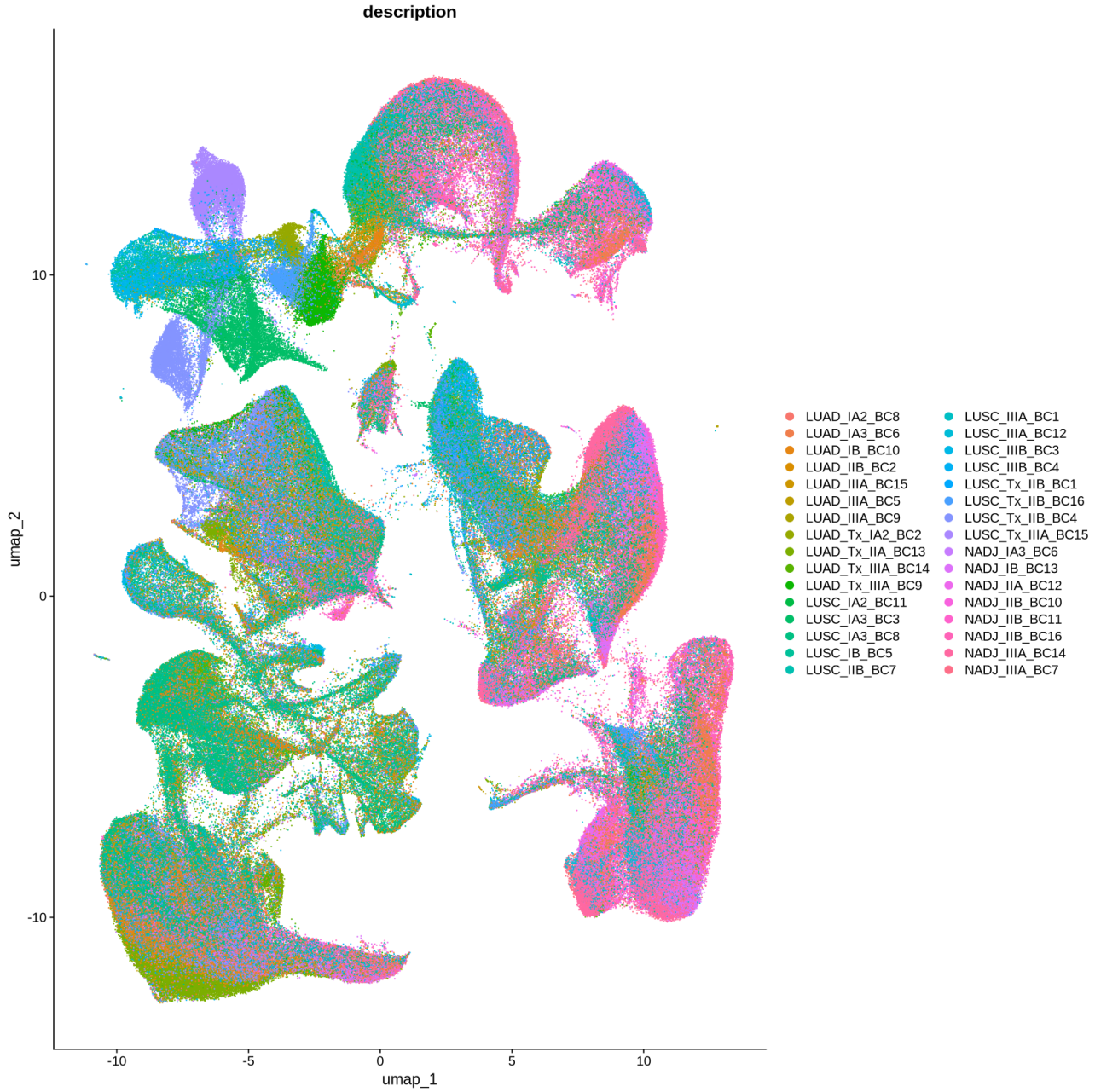
```
################################################################################


work_dir <- args[1]
### WORKING/OUTPUT DIRECTORY
filename <- paste('scrnaseq_sctransform_to_umap_231127', system("date '+_%Y
_%m_%d_%H_%M_%S'", intern=TRUE), '.rds', sep = '')   ### OUTPUT FILENAME

# load packages
library(BPCells)
library(Seurat)
library(SeuratObject)
library(SeuratDisk)
library(Azimuth)
library(future)
library(dplyr)

options(future.globals.maxSize = 16e+09) # 16G

# change the current plan to access parallelization
plan("multisession", workers = availableCores())
plan()

# load data
cat(paste('\n', system("date", intern=TRUE), '---> loading data...', '\n'))
setwd(work_dir)
filename <- "nsclc_2023_11_27_08_03_25.rds"
nsclc <- readRDS(filename)
cat(paste('\n', system("date", intern=TRUE), '---> loading data COMPLETE.',
'\n'))

# normalize & cluster
cat(paste('\n', system("date", intern=TRUE), '---> starting sctransform to u
map...', '\n'))
nsclc <- SCTransform(nsclc, ncells = 100000, conserve.memory = TRUE) %>%
            RunPCA() %>%
                FindNeighbors(dims = 1:30) %>%
                    FindClusters() %>%
                        RunUMAP(dims = 1:30)
cat(paste('\n', system("date", intern=TRUE), '---> sctransform to umap COMPL
ETE.', '\n'))

# save objects
cat(paste('\n', system("date", intern=TRUE), '---> saving output...', '\n'))
saveRDS(nsclc, file = filename)
cat(paste('\n', system("date", intern=TRUE), '---> save objects COMPLETE.',
'\n'))

# cleanup
```

```
rm(nsclc)
gc()

# Explicitly close multisession workers by switching plan
plan(sequential)
plan()

################################################################################

## print session info ##
cat("\nSession Info below: \n")
sessionInfo()

# clear workspace
rm(list = ls())
gc()
```

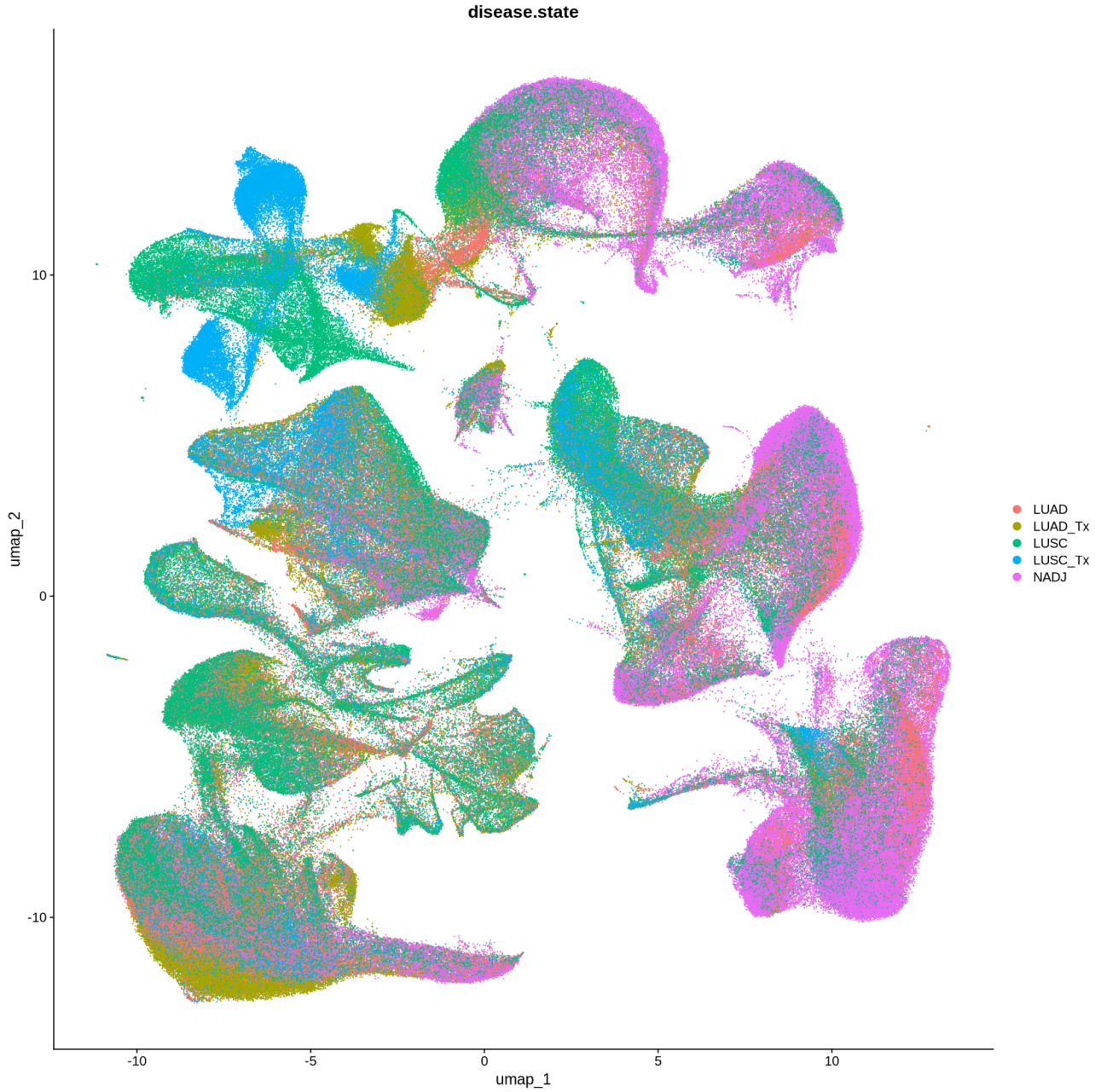
In [10]:
```
# load Seurat object output by SLURM submission
filename <- 'nsclc_2023_11_27_08_03_25.rds'
nsclc <- readRDS(filename)
nsclc
```

```
An object of class Seurat
35488 features across 619727 samples within 2 assays
Active assay: SCT (17451 features, 3000 variable features)
 3 layers present: counts, data, scale.data
 1 other assay present: RNA
 2 dimensional reductions calculated: pca, umap
```

In [11]:
```
# visualize UMAP and group by the 32 samples of origin
DimPlot(nsclc, reduction = "umap",
        group.by = c('description'),
       raster = FALSE)
```



In [12]:
```
# visualize UMAP and group by the 5 biological states
DimPlot(nsclc, reduction = "umap",
        group.by = c('disease.state'),
       raster = FALSE)
```



cell annotation

We then use Azimuth to annotate the cells using the Human Lung Cell Atlas as a reference.

This is also submitted as a SLURM job to the HPC.

In [3]:
```
# R script for Azimuth cell annotation using the Human Lung Cell Atlas as a
file_to_read <- "~/tools/R_scripts/scrnaseq_run_azimuth_231128.r"
file_content <- readLines(file_to_read)
cat(file_content, sep = "\n")
```

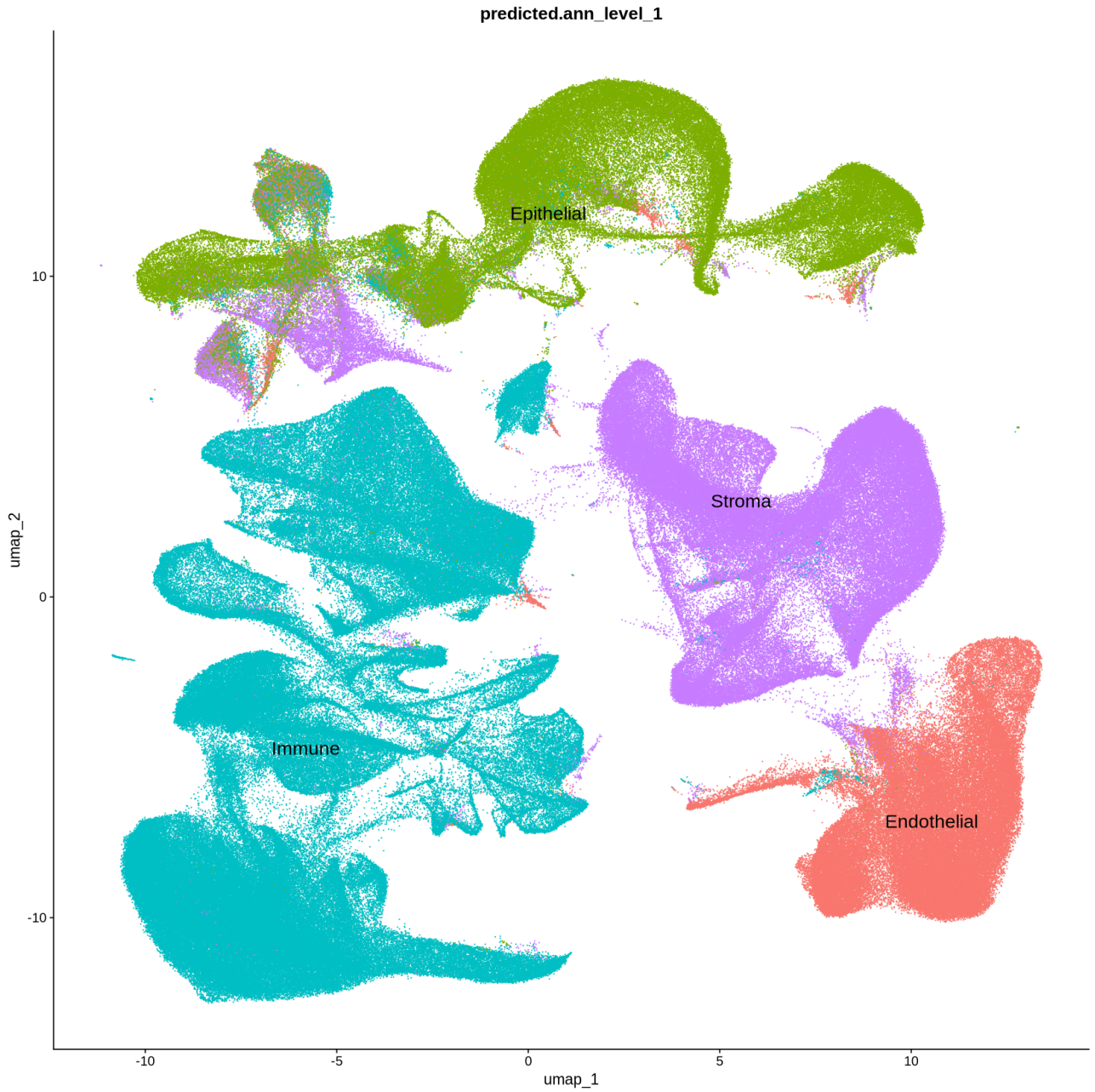
```
# clear workspace
rm(list = ls())
gc()

args <- commandArgs(TRUE) # to access command line arguments
print(args)

#############################################################################


work_dir <- args[1]
### WORKING/OUTPUT DIRECTORY
input_filename <- "nsclc_2023_11_27_08_03_25.rds"
### INPUT FILENAME
output_filename <- paste('scrnaseq_run_azimuth_231128', system("date '+_%Y_%
m_%d_%H_%M_%S'", intern=TRUE), '.rds', sep = '')   ### OUTPUT FILENAME

# load packages
library(BPCells)
library(Seurat)
library(SeuratObject)
library(SeuratDisk)
library(Azimuth)
library(SeuratData)
library(future)
library(dplyr)

options(future.globals.maxSize = 16e+09) # 16G

# change the current plan to access parallelization
plan("multisession", workers = availableCores())
plan()

# load data
cat(paste('\n', system("date", intern=TRUE), '---> loading data...', '\n'))
setwd(work_dir)
nsclc <- readRDS(input_filename)
cat(paste('\n', system("date", intern=TRUE), '---> loading data COMPLETE.',
'\n'))

# annotate with Azimuth
cat(paste('\n', system("date", intern=TRUE), '---> starting annotate with Az
imuth...', '\n'))
options(timeout = 1200)
nsclc <- RunAzimuth(nsclc, reference = "lungref")
cat(paste('\n', system("date", intern=TRUE), '---> annotate with Azimuth COM
PLETE.', '\n'))

# save objects
cat(paste('\n', system("date", intern=TRUE), '---> saving output...', '\n'))
```

```
saveRDS(nsclc, file = output_filename)
cat(paste('\n', system("date", intern=TRUE), '---> save objects COMPLETE.',
'\n'))

# cleanup
rm(nsclc)
gc()

# Explicitly close multisession workers by switching plan
plan(sequential)
plan()

################################################################################

## print session info ##
cat("\nSession Info below: \n")
sessionInfo()

# clear workspace
rm(list = ls())
gc()
```

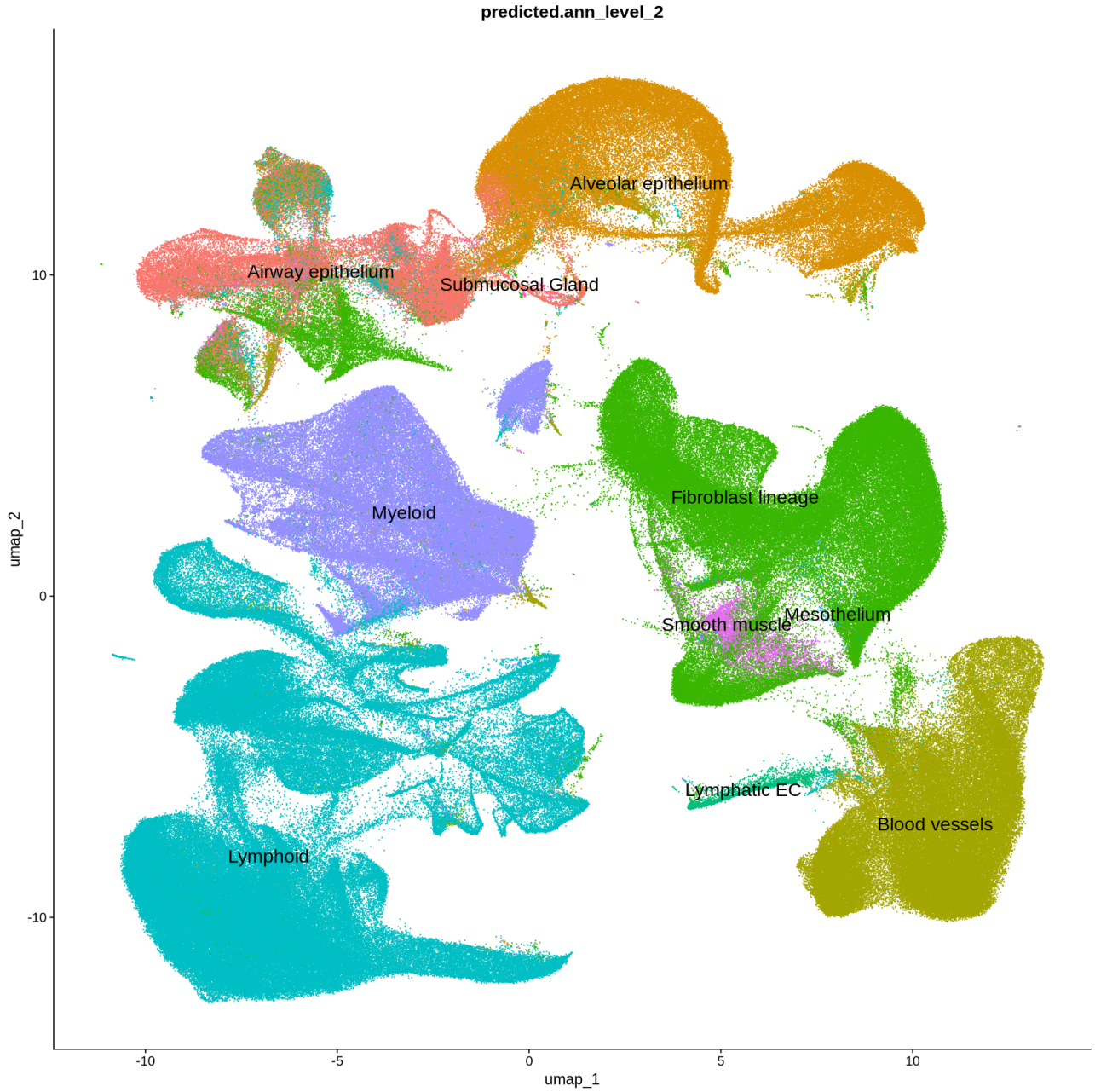
In [4]:
```
# read Azimuth annotated rds
nsclc <- readRDS('scrnaseq_run_azimuth_231128_2023_11_28_11_01_02.rds')
nsclc
```

```
An object of class Seurat
35638 features across 619727 samples within 8 assays
Active assay: SCT (17451 features, 3000 variable features)
 3 layers present: counts, data, scale.data
 7 other assays present: RNA, prediction.score.ann_level_1, prediction.scor
e.ann_level_2, prediction.score.ann_level_3, prediction.score.ann_level_4, p
rediction.score.ann_level_5, prediction.score.ann_finest_level
 4 dimensional reductions calculated: pca, umap, integrated_dr, ref.umap
```

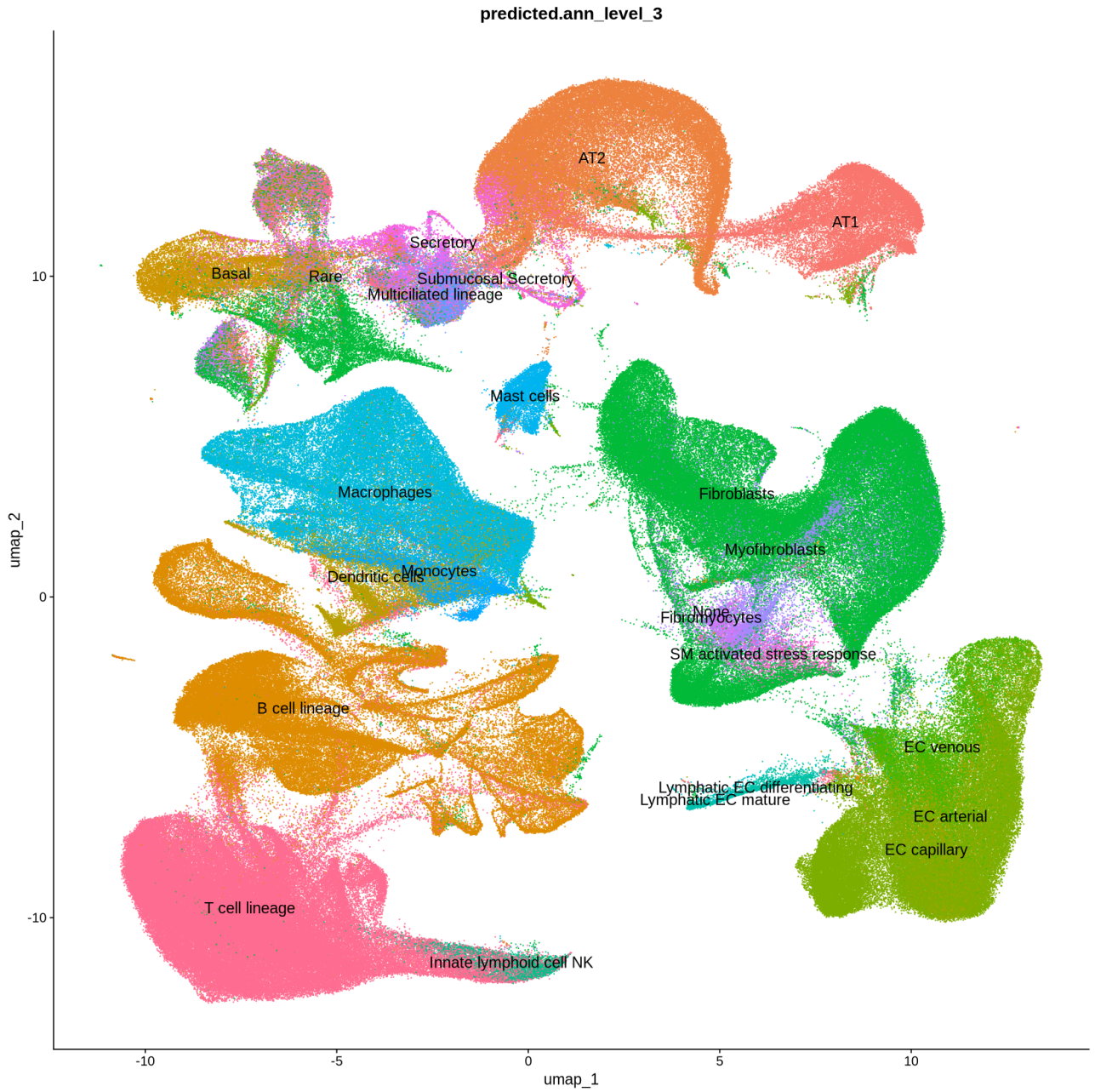
In [5]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_level_1", label = TRUE, label.size
```



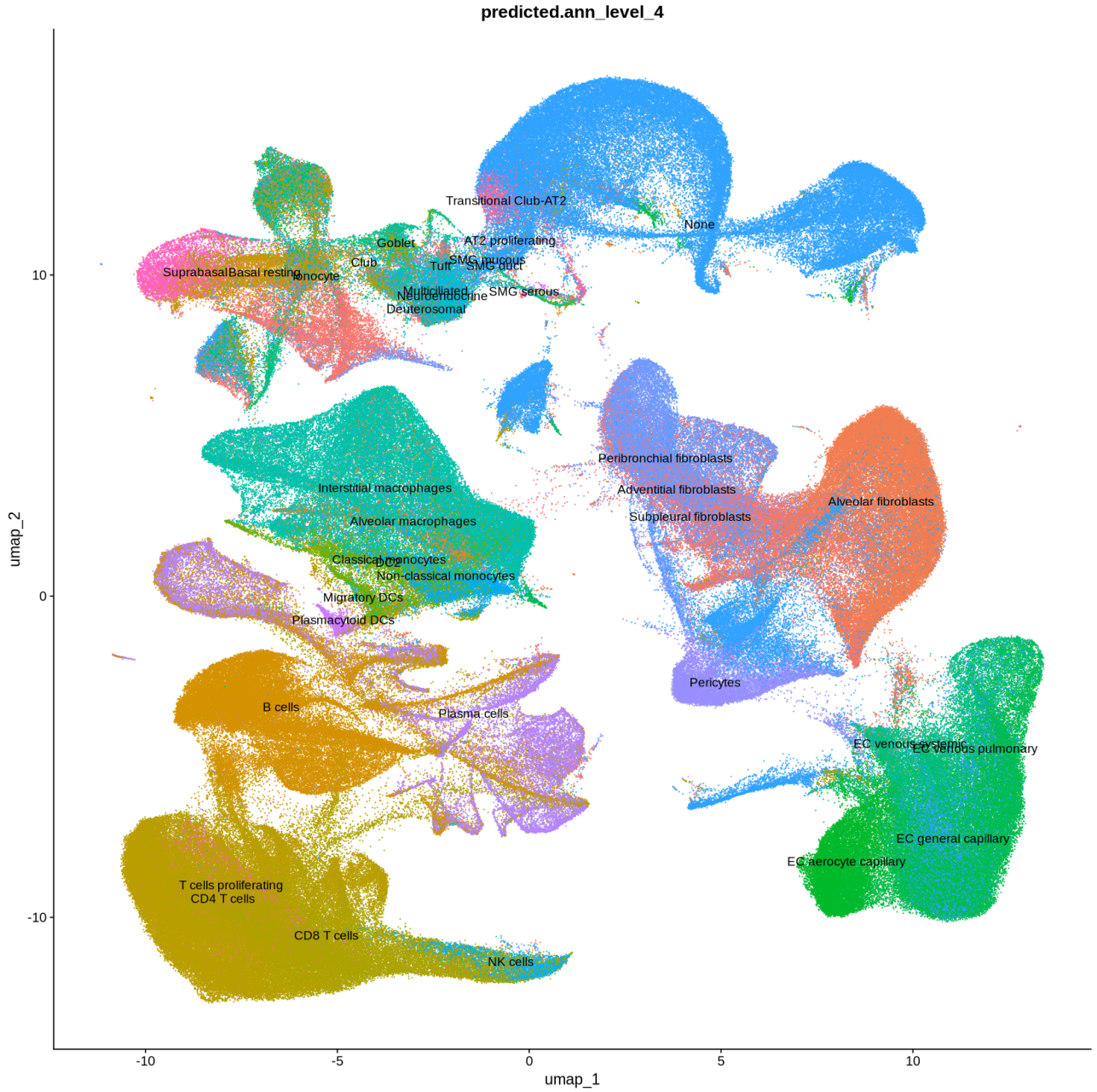
In [6]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_level_2", label = TRUE, label.size
```



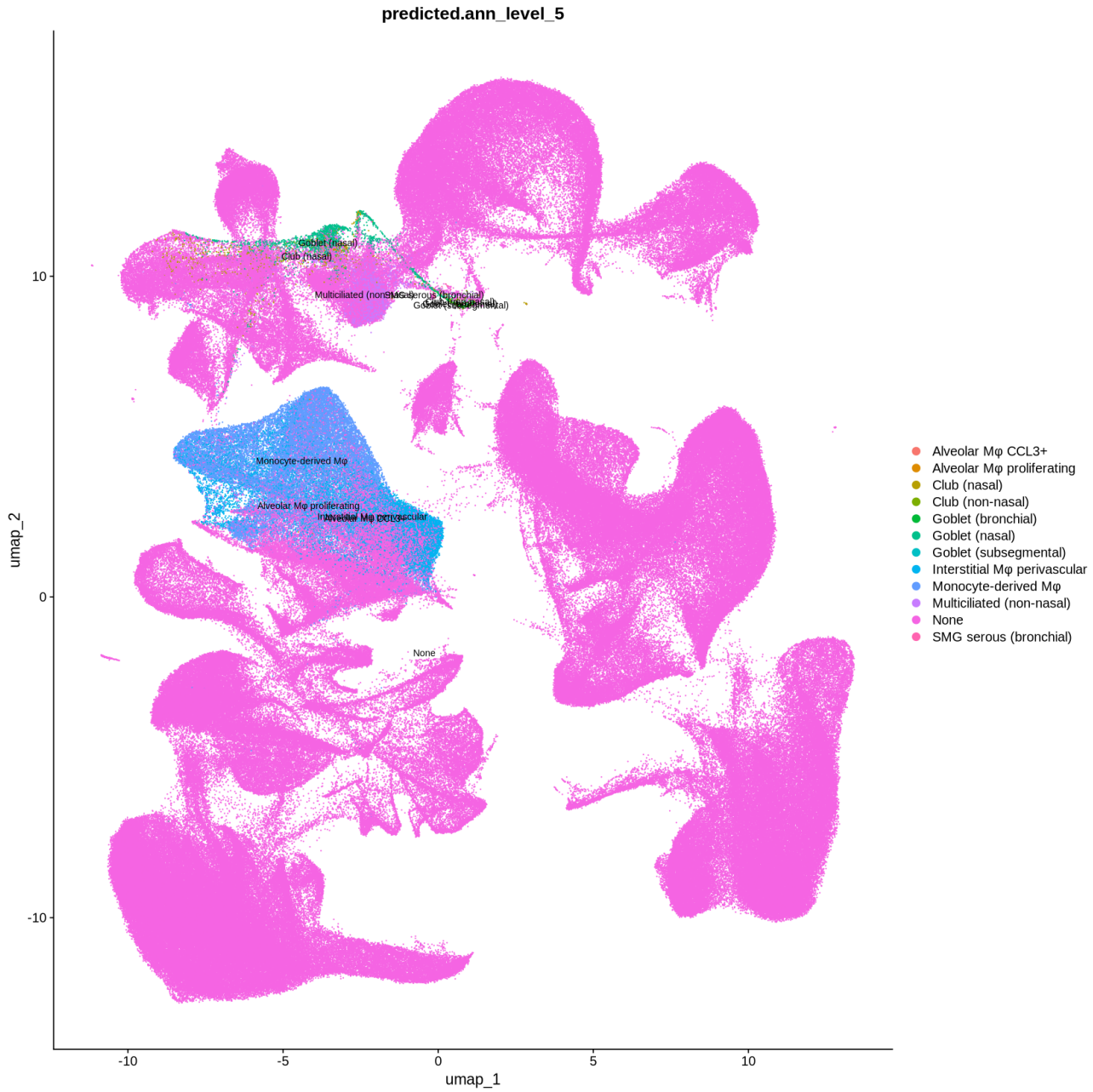
In [7]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_level_3", label = TRUE, label.size
```



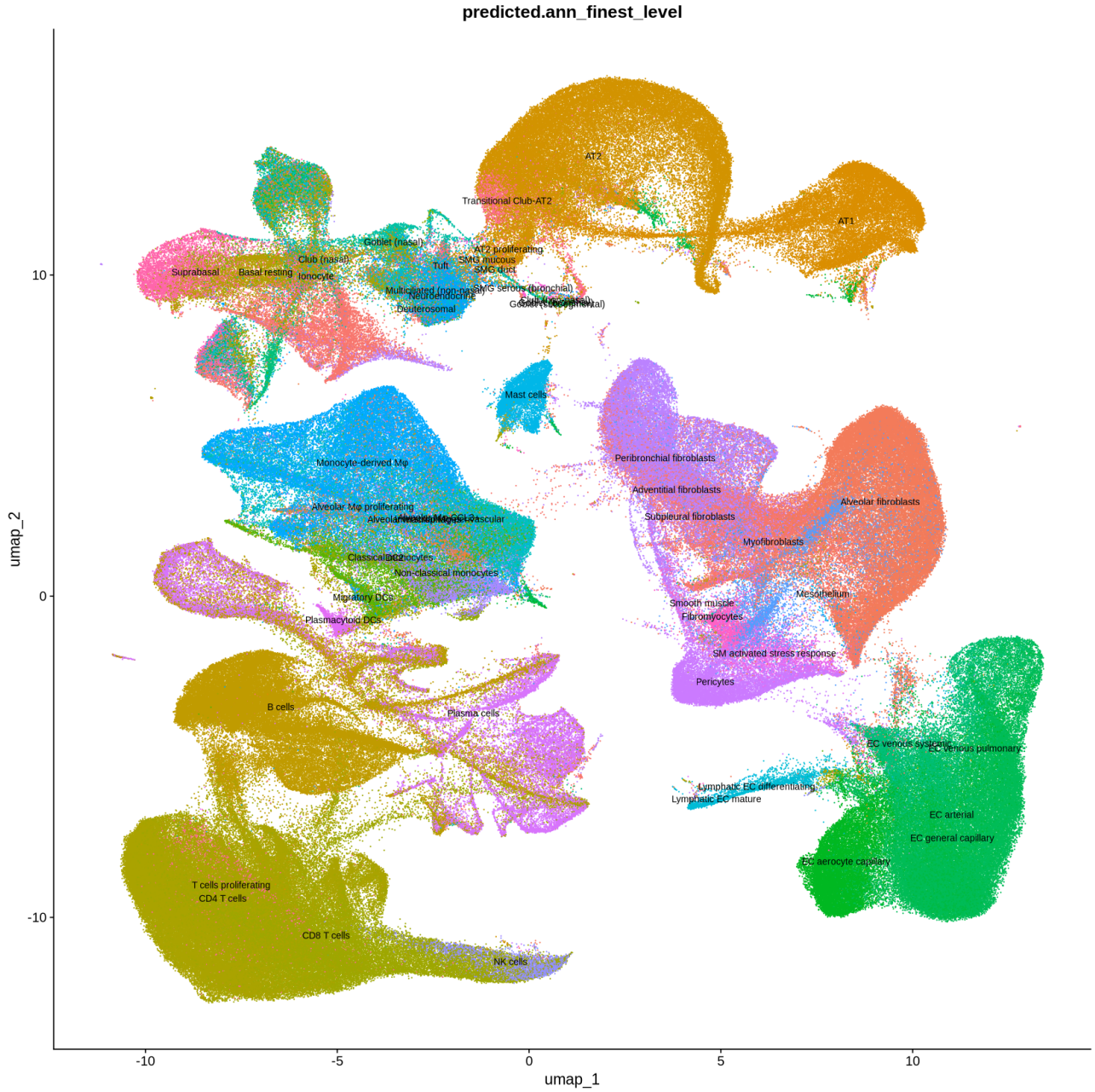
In [8]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_level_4", label = TRUE, label.size
```



In [9]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_level_5", label = TRUE, label.size
```



In [10]:
```
# visualization of Azimuth-annotated data
DimPlot(nsclc, group.by = "predicted.ann_finest_level", label = TRUE, label.
```



pseudobulk analysis

We can then carry out a pseudobulk analysis where we compare all cells by donor or even specific annotated cell types by donor.

In [11]:
```
# pseudobulk by donor
bulk_by_donor <- AggregateExpression(nsclc, return.seurat = TRUE,
                                     assays = "RNA",
                                     group.by = "description")
Cells(bulk_by_donor)
```

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

'LUAD-IA2-BC8' · 'LUAD-IA3-BC6' · 'LUAD-IB-BC10' · 'LUAD-IIB-BC2' · 'LUAD-IIIA-BC15' · 'LUAD-IIIA-BC5' · 'LUAD-IIIA-BC9' · 'LUAD-Tx-IA2-BC2' · 'LUAD-Tx-IIA-BC13' · 'LUAD-Tx-IIIA-BC14' · 'LUAD-Tx-IIIA-BC9' · 'LUSC-IA2-BC11' · 'LUSC-IA3-BC3' · 'LUSC-IA3-BC8' · 'LUSC-IB-BC5' · 'LUSC-IIB-BC7' · 'LUSC-IIIA-BC1' · 'LUSC-IIIA-BC12' · 'LUSC-IIIB-BC3' · 'LUSC-IIIB-BC4' · 'LUSC-Tx-IIB-BC1' · 'LUSC-Tx-IIB-BC16' · 'LUSC-Tx-IIB-BC4' · 'LUSC-Tx-IIIA-BC15' · 'NADJ-IA3-BC6' · 'NADJ-IB-BC13' · 'NADJ-IIA-BC12' · 'NADJ-IIB-BC10' · 'NADJ-IIB-BC11' · 'NADJ-IIB-BC16' · 'NADJ-IIIA-BC14' · 'NADJ-IIIA-BC7'

In [12]:
```r
# extract the five biological groups (conditions = LUAD LUAD-Tx LUSC LUSC-Tx
sample_ids <- Cells(bulk_by_donor)

# Function to extract the desired parts
extract_parts <- function(input) {
  # Split the string by '-'
  parts <- strsplit(input, "-")[[1]]

  # Extract the parts based on positions
  after_last_hyphen <- tail(parts, 1)
  after_second_last_hyphen <- tail(head(parts, -1), 1)
  remainder <- paste(head(parts, -2), collapse = "-")

  return(list(after_last_hyphen, after_second_last_hyphen, remainder))
}

# Apply the function to the input vector
result <- lapply(sample_ids, extract_parts)

# Extracting the three resulting vectors
after_last_hyphen <- sapply(result, `[[`, 1)
after_second_last_hyphen <- sapply(result, `[[`, 2)
condition <- sapply(result, `[[`, 3)

print(condition)
```

```
 [1] "LUAD"    "LUAD"    "LUAD"    "LUAD"    "LUAD"    "LUAD"    "LUAD"
 [8] "LUAD-Tx" "LUAD-Tx" "LUAD-Tx" "LUAD-Tx" "LUSC"    "LUSC"    "LUSC"
[15] "LUSC"    "LUSC"    "LUSC"    "LUSC"    "LUSC"    "LUSC"    "LUSC-Tx"
[22] "LUSC-Tx" "LUSC-Tx" "LUSC-Tx" "NADJ"    "NADJ"    "NADJ"    "NADJ"
[29] "NADJ"    "NADJ"    "NADJ"    "NADJ"
```

In [13]:
```r
# create coldata for DESeq dataset
coldata <- data.frame(condition = as.factor(condition),
                      row.names = colnames(bulk_by_donor[['RNA']]$counts))
```

```
coldata
```

A data.frame: 32 × 1

	condition
	<fct>
LUAD-IA2-BC8	LUAD
LUAD-IA3-BC6	LUAD
LUAD-IB-BC10	LUAD
LUAD-IIB-BC2	LUAD
LUAD-IIIA-BC15	LUAD
LUAD-IIIA-BC5	LUAD
LUAD-IIIA-BC9	LUAD
LUAD-Tx-IA2-BC2	LUAD-Tx
LUAD-Tx-IIA-BC13	LUAD-Tx
LUAD-Tx-IIIA-BC14	LUAD-Tx
LUAD-Tx-IIIA-BC9	LUAD-Tx
LUSC-IA2-BC11	LUSC
LUSC-IA3-BC3	LUSC
LUSC-IA3-BC8	LUSC
LUSC-IB-BC5	LUSC
LUSC-IIB-BC7	LUSC
LUSC-IIIA-BC1	LUSC
LUSC-IIIA-BC12	LUSC
LUSC-IIIB-BC3	LUSC
LUSC-IIIB-BC4	LUSC
LUSC-Tx-IIB-BC1	LUSC-Tx
LUSC-Tx-IIB-BC16	LUSC-Tx
LUSC-Tx-IIB-BC4	LUSC-Tx
LUSC-Tx-IIIA-BC15	LUSC-Tx
NADJ-IA3-BC6	NADJ
NADJ-IB-BC13	NADJ

NADJ-IIA-BC12	NADJ
NADJ-IIB-BC10	NADJ
NADJ-IIB-BC11	NADJ
NADJ-IIB-BC16	NADJ
NADJ-IIIA-BC14	NADJ
NADJ-IIIA-BC7	NADJ

In [14]:
```
# create DESeq object
countdata <- bulk_by_donor[['RNA']]$counts
dds <- DESeqDataSetFromMatrix(countData = countdata,
                              colData = coldata,
                              design = ~ condition)
dds
```

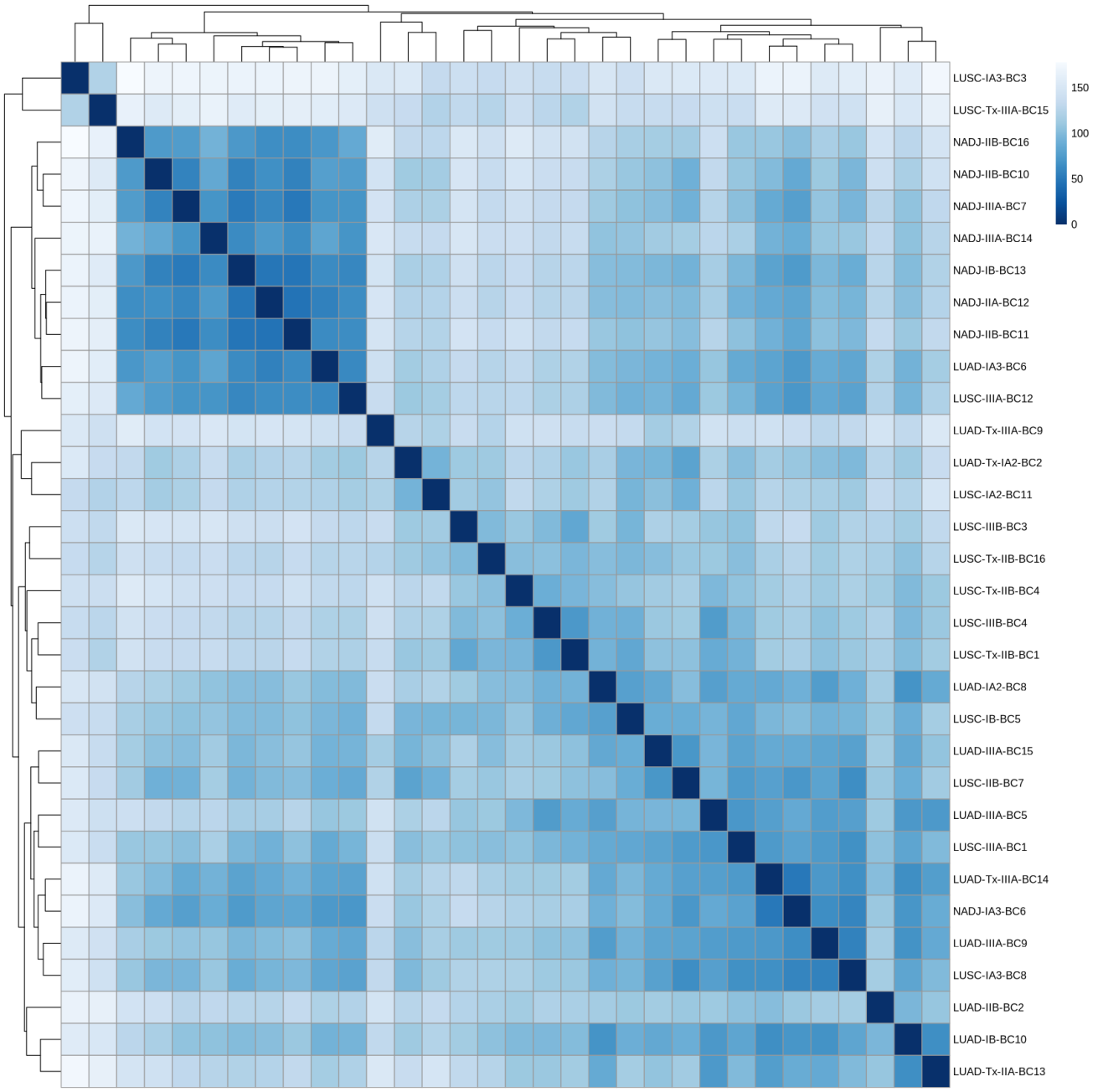
```
converting counts to integer mode

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]

class: DESeqDataSet
dim: 18037 32
metadata(1): version
assays(1): counts
rownames(18037): SAMD11 NOC2L ... MT-ND6 MT-CYB
rowData names(0):
colnames(32): LUAD-IA2-BC8 LUAD-IA3-BC6 ... NADJ-IIIA-BC14
  NADJ-IIIA-BC7
colData names(1): condition
```

In [15]:
```
# drop non-informative (low/no count) genes
keep <- rowSums(counts(dds) >= 10) >= 4 # at least 4 samples (columns) with
dds <- dds[keep, ]
dds
```

```
class: DESeqDataSet
dim: 16303 32
metadata(1): version
assays(1): counts
rownames(16303): SAMD11 NOC2L ... MT-ND6 MT-CYB
rowData names(0):
colnames(32): LUAD-IA2-BC8 LUAD-IA3-BC6 ... NADJ-IIIA-BC14
  NADJ-IIIA-BC7
colData names(1): condition
```

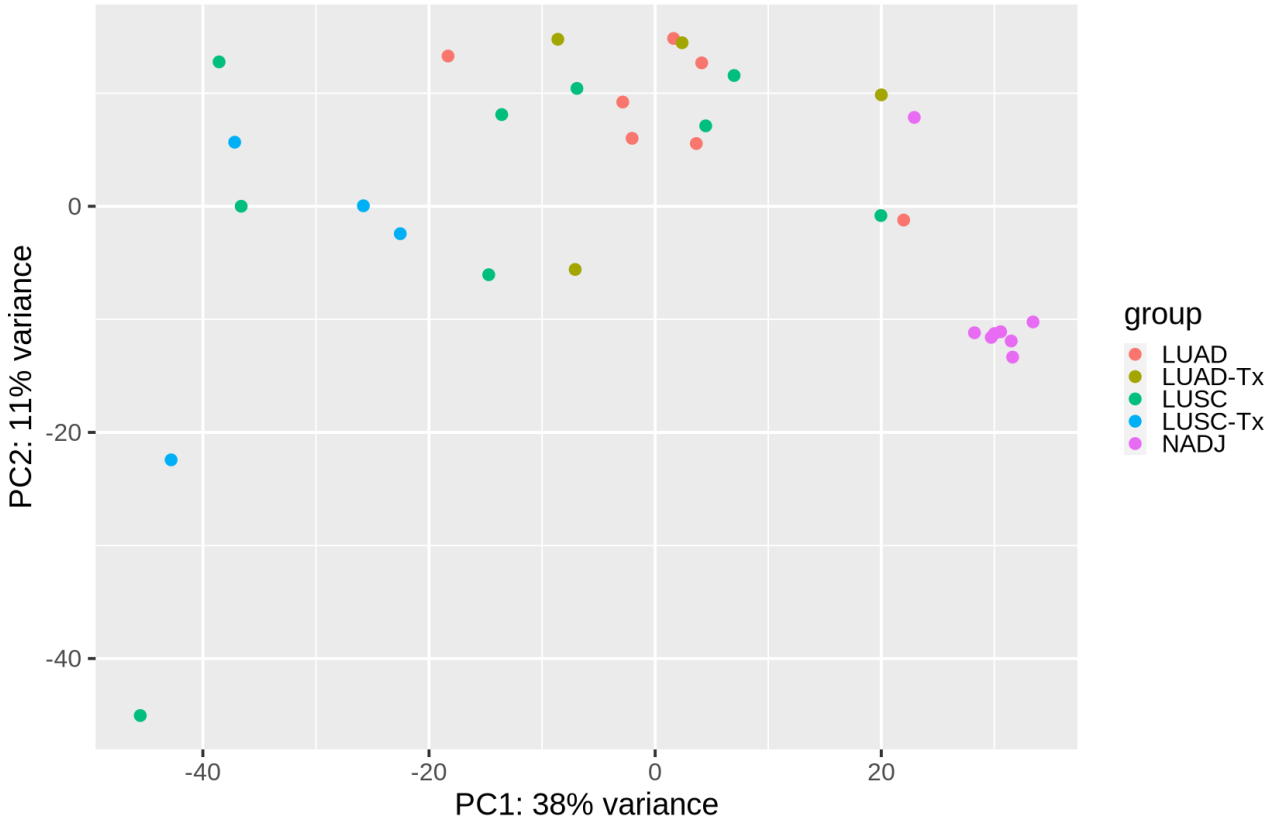
In [16]:
```r
# heatmap of sample distances
rld <- rlog(dds)

sampleDists <- dist(t(assay(rld)))

sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- colnames(rld)
colnames(sampleDistMatrix) <- NULL # rownames(sampleDistMatrix)#NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sampleDists,
         clustering_distance_cols = sampleDists,
         col = colors)
```

```
rlog() may take a few minutes with 30 or more samples,
vst() is a much faster transformation

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]

Found more than one class "dist" in cache; using the first, from namespace '
spam'

Also defined by 'BiocGenerics'

Found more than one class "dist" in cache; using the first, from namespace '
spam'

Also defined by 'BiocGenerics'
```



In [17]:
```
# PCA plot
object <- rld
ntop <- 500 # number of variable genes to use for PCA
intgroup <- 'condition'
pcsToUse = 1:2

rv <- rowVars(assay(object))
select <- order(rv, decreasing = TRUE)[seq_len(min(ntop, length(rv)))]
pca <- prcomp(t(assay(object)[select, ]))
percentVar <- pca$sdev^2/sum(pca$sdev^2)

intgroup.df <- as.data.frame(colData(object)[, intgroup, drop = FALSE])
group <- colData(object)[[intgroup]]
```

```
pcs <- paste0("PC", pcsToUse)
d <- data.frame(V1 = pca$x[, pcsToUse[1]], V2 = pca$x[, pcsToUse[2]],
                group = group, intgroup.df, name = colnames(object))
colnames(d)[1:2] <- pcs

ggplot(data = d, aes(x = PC1, y = PC2, color = group)) +
    geom_point(size = 4) +
    xlab(paste0(pcs[1], ": ", round(percentVar[pcsToUse[1]] * 100), "% varia
    ylab(paste0(pcs[2], ": ", round(percentVar[pcsToUse[2]] * 100), "% varia
    coord_fixed() +
    theme_gray(base_size = 24)
```



Other than the clustering together of the NADJ cells, no clear patterns emerge from either the heatmap or the PCA plot.

differential expression analysis

In [18]:
```
# differential expression analysis
dds <- DESeq(dds)
dds
```

```
estimating size factors

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]

final dispersion estimates

fitting model and testing

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]

-- replacing outliers and refitting for 226 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)

estimating dispersions

fitting model and testing

  Note: levels of factors in the design contain characters other than
  letters, numbers, '_' and '.'. It is recommended (but not required) to use
  only letters, numbers, and delimiters '_' or '.', as these are safe charac
ters
  for column names in R. [This is a message, not a warning or an error]
```

```
class: DESeqDataSet
dim: 16303 32
metadata(1): version
assays(6): counts mu ... replaceCounts replaceCooks
rownames(16303): SAMD11 NOC2L ... MT-ND6 MT-CYB
rowData names(35): baseMean baseVar ... maxCooks replace
colnames(32): LUAD-IA2-BC8 LUAD-IA3-BC6 ... NADJ-IIIA-BC14
  NADJ-IIIA-BC7
colData names(3): condition sizeFactor replaceable
```

In [48]:
```r
res <- results(dds)
summary(res)
res
```

```
out of 16303 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 2033, 12%
LFC < 0 (down)     : 2507, 15%
outliers [1]       : 137, 0.84%
low counts [2]     : 0, 0%
(mean count < 2)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

log2 fold change (MLE): condition NADJ vs LUAD
Wald test p-value: condition NADJ vs LUAD
DataFrame with 16303 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat      pvalue        padj
        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
SAMD11   433.7955      -0.681523  0.503806 -1.352749 0.176135900  0.33773134
NOC2L   2279.7745       0.193390  0.224719  0.860586 0.389466366  0.56711523
KLHL17  1099.8950      -0.210579  0.348688 -0.603919 0.545897563  0.70529534
PLEKHN1  888.0957      -1.219242  0.489036 -2.493151 0.012661507  0.05849117
PERM1     39.2272       2.005662  0.596716  3.361164 0.000776148  0.00859987
...           ...            ...       ...       ...         ...         ...
MT-ND4L  14210.94       -1.46441  0.323998  -4.51981 6.18951e-06 3.07640e-04
MT-ND4   26598.11       -2.84193  0.493873  -5.75438 8.69607e-09 3.34716e-06
MT-ND5    6073.39       -2.30849  0.503327  -4.58646 4.50835e-06 2.50454e-04
MT-ND6   70577.62       -2.05887  0.501072  -4.10894 3.97485e-05 1.15046e-03
MT-CYB   11746.32       -2.31759  0.492815  -4.70276 2.56665e-06 1.74338e-04
```

In [63]:
```r
# using more stringent thresholds: BH FDR adjusted p-value 0.05 vs 0.1
res <- results(dds, alpha = 0.05)
summary(res)
res
```

```
out of 16303 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1415, 8.7%
LFC < 0 (down)     : 1832, 11%
outliers [1]       : 137, 0.84%
low counts [2]     : 0, 0%
(mean count < 2)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

log2 fold change (MLE): condition NADJ vs LUAD
Wald test p-value: condition NADJ vs LUAD
DataFrame with 16303 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat      pvalue        padj
        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
SAMD11   433.7955      -0.681523  0.503806 -1.352749 0.176135900  0.33773134
NOC2L   2279.7745       0.193390  0.224719  0.860586 0.389466366  0.56711523
KLHL17  1099.8950      -0.210579  0.348688 -0.603919 0.545897563  0.70529534
PLEKHN1  888.0957      -1.219242  0.489036 -2.493151 0.012661507  0.05849117
PERM1     39.2272       2.005662  0.596716  3.361164 0.000776148  0.00859987
...           ...            ...       ...       ...         ...         ...
MT-ND4L  14210.94       -1.46441  0.323998  -4.51981 6.18951e-06 3.07640e-04
MT-ND4   26598.11       -2.84193  0.493873  -5.75438 8.69607e-09 3.34716e-06
MT-ND5    6073.39       -2.30849  0.503327  -4.58646 4.50835e-06 2.50454e-04
MT-ND6   70577.62       -2.05887  0.501072  -4.10894 3.97485e-05 1.15046e-03
MT-CYB   11746.32       -2.31759  0.492815  -4.70276 2.56665e-06 1.74338e-04
```

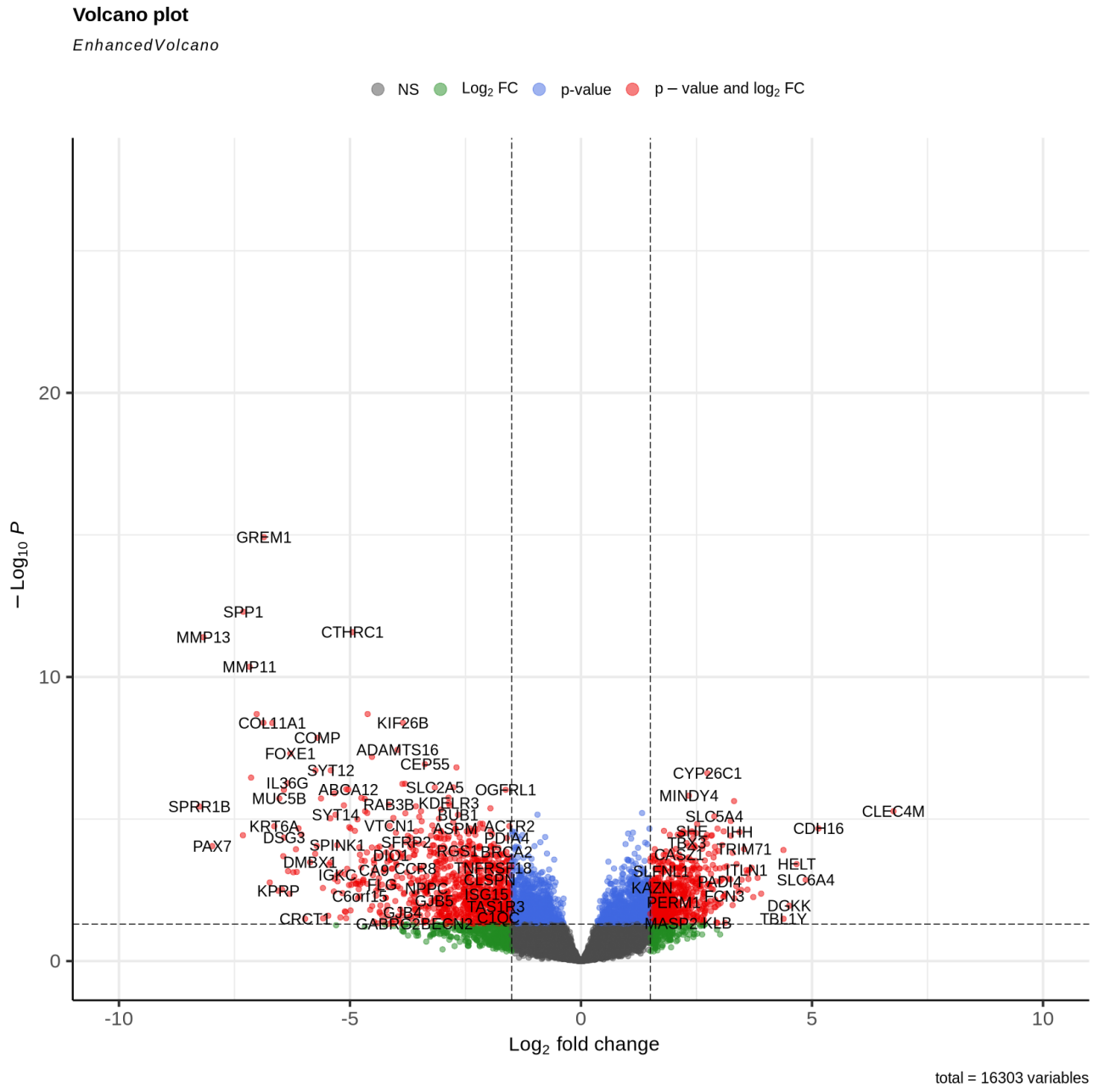
In [64]:
```
# significant genes with strongest down-regulation in NADJ
resSig <- subset(res, padj < 0.05)
head(resSig[ order(resSig$log2FoldChange), ])
```

```
log2 fold change (MLE): condition NADJ vs LUAD
Wald test p-value: condition NADJ vs LUAD
DataFrame with 6 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat      pvalue        pad
j
        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric
>
SPRR2E     8.35502      -25.46935  2.342740 -10.87161 1.57406e-27 2.54462e-2
3
SPRR1B   458.47772       -8.25600  1.440096  -5.73295 9.86973e-09 3.62623e-0
6
MMP13    790.70214       -8.17280  1.021410  -8.00149 1.22920e-15 3.97423e-1
2
PAX7      44.88256       -7.98151  1.629808  -4.89721 9.72068e-07 8.97596e-0
5
A2ML1    138.46412       -7.31625  1.424191  -5.13713 2.78971e-07 3.72715e-0
5
SPP1   22935.70228       -7.30798  0.879252  -8.31159 9.44268e-17 5.08835e-1
3
```

In [65]:
```
# significant genes with strongest up-regulation in NADJ
head(resSig[ order(resSig$log2FoldChange, decreasing = TRUE), ])
```

```
log2 fold change (MLE): condition NADJ vs LUAD
Wald test p-value: condition NADJ vs LUAD
DataFrame with 6 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat      pvalue        padj
        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
CLEC4M  166.62403        6.76344   1.19719   5.64943 1.60985e-08 5.31120e-06
CDH16    18.77741        5.14150   0.96595   5.32274 1.02219e-07 2.14606e-05
SLC6A4 1602.14551        4.86331   1.20405   4.03911 5.36535e-05 1.38778e-03
HELT     11.56360        4.66740   1.04903   4.44926 8.61662e-06 3.84796e-04
DGKK      5.74979        4.50814   1.38622   3.25212 1.14548e-03 1.12161e-02
TBL1Y   104.09172        4.38477   1.57359   2.78648 5.32842e-03 3.25144e-02
```

In [66]:
```
EnhancedVolcano(as.data.frame(res), x = 'log2FoldChange', y = 'padj', lab =
                pCutoff = 0.05, FCcutoff = 1.5,
                xlim = c(-10, 10))
```



In [67]:
```
res[c('AGER', 'EMP2'),]
```

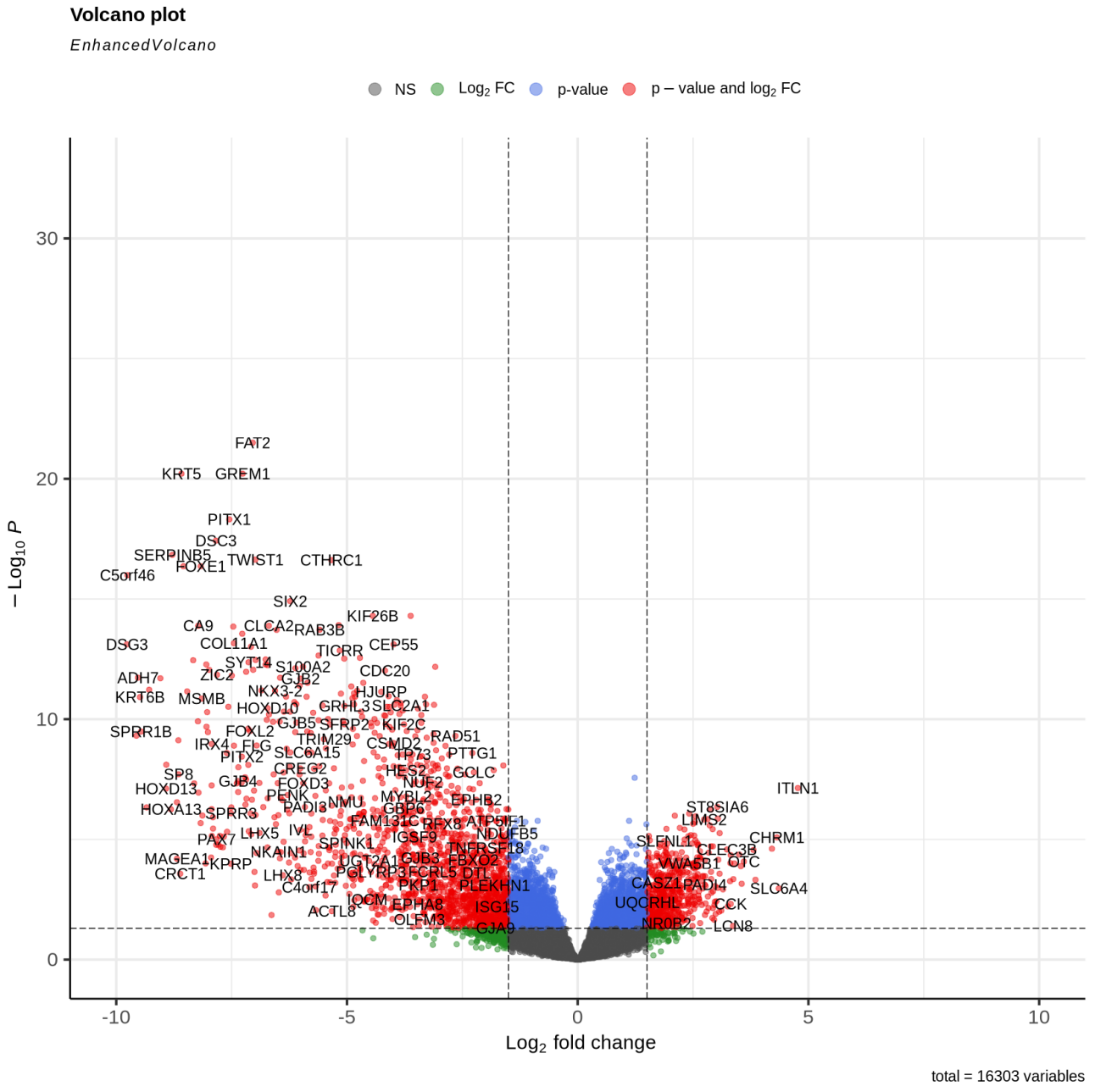
```
log2 fold change (MLE): condition NADJ vs LUAD
Wald test p-value: condition NADJ vs LUAD
DataFrame with 2 rows and 6 columns
      baseMean log2FoldChange     lfcSE      stat     pvalue      padj
     <numeric>      <numeric> <numeric> <numeric>  <numeric> <numeric>
AGER   12247.6        1.70476  0.621273   2.74397 0.00607004 0.0355666
EMP2   10128.3        1.79480  0.650439   2.75937 0.00579133 0.0344581
```

In [68]:
```
# compare NADJ to LUSC
res <- results(dds, alpha = 0.05, contrast = c("condition", "NADJ", "LUSC"))
summary(res)
```

```
res
```

```
out of 16303 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1660, 10%
LFC < 0 (down)     : 2926, 18%
outliers [1]       : 137, 0.84%
low counts [2]     : 0, 0%
(mean count < 2)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

log2 fold change (MLE): condition NADJ vs LUSC
Wald test p-value: condition NADJ vs LUSC
DataFrame with 16303 rows and 6 columns
         baseMean log2FoldChange     lfcSE      stat      pvalue        padj
        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
SAMD11   433.7955      -0.364878  0.466774 -0.781703 4.34389e-01 0.594962141
NOC2L   2279.7745      -0.336687  0.206945 -1.626945 1.03749e-01 0.219380056
KLHL17  1099.8950      -0.844994  0.321651 -2.627048 8.61293e-03 0.034387895
PLEKHN1  888.0957      -1.802053  0.453349 -3.974983 7.03844e-05 0.000820948
PERM1     39.2272       1.242017  0.540102  2.299598 2.14710e-02 0.068125660
...           ...            ...       ...       ...         ...         ...
MT-ND4L  14210.94      -0.841548  0.303738 -2.770637 5.59468e-03 2.47249e-02
MT-ND4   26598.11      -1.217122  0.463596 -2.625392 8.65493e-03 3.45045e-02
MT-ND5    6073.39      -0.453398  0.472204 -0.960175 3.36967e-01 5.01752e-01
MT-ND6   70577.62      -2.837879  0.470353 -6.033509 1.60437e-09 1.01313e-07
MT-CYB   11746.32      -0.892440  0.462467 -1.929739 5.36392e-02 1.35362e-01
```

In [69]:
```
EnhancedVolcano(as.data.frame(res), x = 'log2FoldChange', y = 'padj', lab =
                pCutoff = 0.05, FCcutoff = 1.5,
                xlim = c(-10, 10))
```



In [70]:
```
res[c('AGER', 'EMP2'),]
```

```
log2 fold change (MLE): condition NADJ vs LUSC
Wald test p-value: condition NADJ vs LUSC
DataFrame with 2 rows and 6 columns
      baseMean log2FoldChange     lfcSE      stat      pvalue        padj
     <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
AGER   12247.6        1.93027  0.582796   3.31208 9.26054e-04 0.006094182
EMP2   10128.3        2.56706  0.610142   4.20732 2.58417e-05 0.000363583
```

We see that the tumor suppressor genes AGER and EMP2 are upregulated in normal

adjacent cells compared to both lung adenocarcinoma and lung squamous cell cancer.

NEXT:

- pseudobulk analysis of AT1 and AT2 cells which are thought to be the cells of origin of lung adenocarcinoma
- gene set enrichment analysis (GSEA) comparing the 5 biological conditions

In [29]:
```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: x86_64-conda-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)

Matrix products: default
BLAS/LAPACK: /home/fmbuga/.conda/envs/seurat5/lib/libopenblasp-r0.3.24.so;
LAPACK version 3.11.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: US/Pacific
tzcode source: system (glibc)

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] ggplot2_3.4.4               RColorBrewer_1.1-3
 [3] pheatmap_1.0.12             DESeq2_1.42.0
 [5] SummarizedExperiment_1.32.0 Biobase_2.62.0
 [7] MatrixGenerics_1.14.0       matrixStats_1.1.0
 [9] GenomicRanges_1.54.1        GenomeInfoDb_1.38.1
[11] IRanges_2.36.0              S4Vectors_0.40.1
[13] BiocGenerics_0.48.1         future_1.33.0
[15] Azimuth_0.5.0               shinyBS_0.61.1
[17] BPCells_0.1.0               SeuratDisk_0.0.0.9021
[19] Seurat_5.0.1                SeuratObject_5.0.1
[21] sp_2.1-1                    dplyr_1.1.4

loaded via a namespace (and not attached):
  [1] fs_1.6.3                          ProtGenerics_1.34.0
```

```
  [3] spatstat.sparse_3.0-3           bitops_1.0-7
  [5] DirichletMultinomial_1.44.0     TFBSTools_1.40.0
  [7] httr_1.4.7                      repr_1.1.6
  [9] tools_4.3.2                     sctransform_0.4.1
 [11] utf8_1.2.4                      R6_2.5.1
 [13] DT_0.30                         lazyeval_0.2.2
 [15] uwot_0.1.16                     rhdf5filters_1.14.1
 [17] withr_2.5.2                     prettyunits_1.2.0
 [19] gridExtra_2.3                   progressr_0.14.0
 [21] cli_3.6.1                       spatstat.explore_3.2-5
 [23] fastDummies_1.7.3               EnsDb.Hsapiens.v86_2.99.0
 [25] shinyjs_2.1.0                   labeling_0.4.3
 [27] spatstat.data_3.0-3             readr_2.1.4
 [29] ggridges_0.5.4                  pbapply_1.7-2
 [31] Rsamtools_2.18.0                pbdZMQ_0.3-10
 [33] R.utils_2.12.3                  parallelly_1.36.0
 [35] BSgenome_1.70.1                 RSQLite_2.3.3
 [37] generics_0.1.3                  BiocIO_1.12.0
 [39] gtools_3.9.5                    ica_1.0-3
 [41] spatstat.random_3.2-1           googlesheets4_1.1.1
 [43] GO.db_3.18.0                    Matrix_1.6-3
 [45] fansi_1.0.5                     abind_1.4-5
 [47] R.methodsS3_1.8.2               lifecycle_1.0.4
 [49] yaml_2.3.7                      rhdf5_2.46.0
 [51] SparseArray_1.2.2               BiocFileCache_2.10.1
 [53] Rtsne_0.16                      grid_4.3.2
 [55] blob_1.2.4                      promises_1.2.1
 [57] shinydashboard_0.7.2            crayon_1.5.2
 [59] miniUI_0.1.1.1                  lattice_0.22-5
 [61] cowplot_1.1.1                   GenomicFeatures_1.54.1
 [63] annotate_1.80.0                 KEGGREST_1.42.0
 [65] pillar_1.9.0                    rjson_0.2.21
 [67] future.apply_1.11.0             codetools_0.2-19
 [69] fastmatch_1.1-4                 leiden_0.4.3.1
 [71] glue_1.6.2                      data.table_1.14.8
 [73] vctrs_0.6.4                     png_0.1-8
 [75] spam_2.10-0                     cellranger_1.1.0
 [77] gtable_0.3.4                    poweRlaw_0.70.6
 [79] cachem_1.0.8                    Signac_1.12.0
 [81] S4Arrays_1.2.0                  mime_0.12
 [83] pracma_2.4.4                    survival_3.5-7
 [85] gargle_1.5.2                    RcppRoll_0.3.0
 [87] ellipsis_0.3.2                  fitdistrplus_1.1-11
 [89] ROCR_1.0-11                     nlme_3.1-163
 [91] bit64_4.0.5                     progress_1.2.2
 [93] filelock_1.0.2                  RcppAnnoy_0.0.21
 [95] irlba_2.3.5.1                   KernSmooth_2.23-22
 [97] colorspace_2.1-0                seqLogo_1.68.0
 [99] DBI_1.1.3                       tidyselect_1.2.0
[101] bit_4.0.5                       compiler_4.3.2
```

```
[103] curl_5.1.0                    hdf5r_1.3.8
[105] xml2_1.3.5                    DelayedArray_0.28.0
[107] plotly_4.10.3                 rtracklayer_1.62.0
[109] scales_1.2.1                  caTools_1.18.2
[111] lmtest_0.9-40                 rappdirs_0.3.3
[113] stringr_1.5.1                 digest_0.6.33
[115] goftest_1.2-3                 presto_1.0.0
[117] spatstat.utils_3.0-4          XVector_0.42.0
[119] htmltools_0.5.7               pkgconfig_2.0.3
[121] base64enc_0.1-3               dbplyr_2.4.0
[123] fastmap_1.1.1                 ensembldb_2.26.0
[125] rlang_1.1.2                   htmlwidgets_1.6.3
[127] shiny_1.8.0                   farver_2.1.1
[129] zoo_1.8-12                    jsonlite_1.8.7
[131] BiocParallel_1.36.0           R.oo_1.25.0
[133] RCurl_1.98-1.13               magrittr_2.0.3
[135] GenomeInfoDbData_1.2.11       dotCall64_1.1-0
[137] patchwork_1.1.3               Rhdf5lib_1.24.0
[139] IRkernel_1.3.2                munsell_0.5.0
[141] Rcpp_1.0.11                   reticulate_1.34.0
[143] stringi_1.8.1                 zlibbioc_1.48.0
[145] MASS_7.3-60                   plyr_1.8.9
[147] parallel_4.3.2                listenv_0.9.0
[149] ggrepel_0.9.4                 deldir_1.0-9
[151] CNEr_1.38.0                   Biostrings_2.70.1
[153] IRdisplay_1.1                 splines_4.3.2
[155] tensor_1.5                    hms_1.1.3
[157] locfit_1.5-9.8                BSgenome.Hsapiens.UCSC.hg38_1.4.5
[159] igraph_1.5.1                  uuid_1.1-1
[161] spatstat.geom_3.2-7           RcppHNSW_0.5.0
[163] reshape2_1.4.4                biomaRt_2.58.0
[165] TFMPvalue_0.0.9               XML_3.99-0.15
[167] evaluate_0.23                 JASPAR2020_0.99.10
[169] tzdb_0.4.0                    httpuv_1.6.12
[171] RANN_2.6.1                    tidyr_1.3.0
[173] purrr_1.0.2                   polyclip_1.10-6
[175] SeuratData_0.2.2.9001         scattermore_1.2
[177] xtable_1.8-4                  restfulr_0.0.15
[179] AnnotationFilter_1.26.0       RSpectra_0.16-1
[181] later_1.3.1                   googledrive_2.1.1
[183] viridisLite_0.4.2             tibble_3.2.1
[185] memoise_2.0.1                 AnnotationDbi_1.64.1
[187] GenomicAlignments_1.38.0      cluster_2.1.4
[189] globals_0.16.2
```