Mouse_sorted_lung_cells

Guadalupe Rivera Torruco

9/18/2021

Meta-analysis of different sorted lung populations by: Guadalupe Rivera-Torruco Projects: E-MTAB-8573, E-GEOD-50927, GSE156692, GSE172104, E-GEOD-59831, E-MTAB-10324, E-GEOD-57391, GSE168529 Whole lung cells vs sorted cells RNAseq of selected cells Organism: Mouse C57BL6/J

This analysis correspond to an ongoing project to study different lung subsets from databases and experimental data to analyze Isthmin-1. ISM1 is a secreted protein highly expressed in mouse lungs, also this protein is related to hematopoiesis in zebra fish and in mouse lung progenitors.

References: https://pubmed.ncbi.nlm.nih.gov/29758043/ https://pubmed.ncbi.nlm.nih.gov/35402623/

ENA projects listed were downloaded, quality assesed by FastQc, trimmed and filtered with AfterQC. After quality control, files were quatified with Salmon using the genome assembly GRCm38 from Ensembl. The following script used Salmon Quants (file.sf) and R tools like txtimport and DESeq to annotate and analyse them.

```
options(warn=-1)
```

```
## Bioconductor version '3.12' is out-of-date; the current release version '3.15'
##   is available with R version '4.2'; see https://bioconductor.org/install
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
## corrplot 0.92 loaded
```

If regular installing doesnt work for igraph, try:

install.packages("igraph", type="binary")

If locfit error and R.version <4.1, try or update R: install.packages("http://cran.nexr.com/src/contrib/locfit_1.5-9.1.tar.gz", repos=NULL, type="source")

```
## Bioconductor version 3.12 (BiocManager 1.30.17), R 4.0.5 (2021-03-31)
```

```
## Installing package(s) 'tximportData', 'tximport', 'DESeq2', 'apeglm', 'sva'
```

```
## package 'tximport' successfully unpacked and MD5 sums checked
## package 'DESeq2' successfully unpacked and MD5 sums checked
## package 'apeglm' successfully unpacked and MD5 sums checked
## package 'sva' successfully unpacked and MD5 sums checked
##
```

```
## The downloaded binary packages are in
##  C:\Users\lupit\AppData\Local\Temp\Rtmp69PdR5\downloaded_packages

## installing the source package 'tximportData'

## Old packages: 'cli', 'dplyr', 'ggplot2', 'igraph', 'nloptr', 'tibble'

##
##   There are binary versions available but the source versions are later:
##        binary source needs_compilation
## cli     3.2.0  3.3.0              TRUE
## dplyr   1.0.8  1.0.9              TRUE
## nloptr  2.0.0  2.0.1              TRUE
## tibble  3.1.6  3.1.7              TRUE

## installing the source packages 'cli', 'dplyr', 'nloptr', 'tibble'

## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:igraph':
##
##     normalize, path, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: GenomicRanges
## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'MatrixGenerics'
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Attaching package: 'Biobase'
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
## Loading required package: mgcv
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:IRanges':
##
##     collapse

## This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.

## Loading required package: genefilter

##
## Attaching package: 'genefilter'

## The following objects are masked from 'package:MatrixGenerics':
##
##     rowSds, rowVars

## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars

## Loading required package: BiocParallel
```

```
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.align = "center")
```

```
coldata <- read.csv("metadata.csv", row.names = "Sample", stringsAsFactors=FALSE)
head(coldata)
```

```
##          project_name    ENA_SAMPLE batch Sample_source_name  cell_type
## sample01  E-MTAB-8573  SAMEA6377060     1            WT lung Whole_lung
## sample02  E-MTAB-8573 SAMEA6377061     1            WT lung Whole_lung
## sample03  E-MTAB-8573  SAMEA6377062     1            WT lung Whole_lung
## sample04 E-GEOD-50927  SAMN02358107     2            WT lung Whole_lung
## sample05 E-GEOD-50927  SAMN02358107     2            WT lung Whole_lung
## sample06    GSE156692  SAMN15888043     3            WT lung Whole_lung
##          sample_description disease organism_part     organism
## sample01 lung cell unsorted    none          lung Mus musculus
## sample02 lung cell unsorted    none          lung Mus musculus
## sample03 lung cell unsorted    none          lung Mus musculus
## sample04 lung cell unsorted    none          lung Mus musculus
## sample05 lung cell unsorted    none          lung Mus musculus
## sample06 lung cell unsorted    none          lung Mus musculus
##                        strain age..weeks.     Protocol.REF Term.Source.REF
## sample01              C57BL/6      16-Aug     P-MTAB-91699    ArrayExpress
## sample02              C57BL/6      16-Aug     P-MTAB-91699    ArrayExpress
## sample03              C57BL/6      16-Aug     P-MTAB-91699    ArrayExpress
## sample04 Mixed C57BL/6+others           8     P-GSE50927-3    ArrayExpress
## sample05 Mixed C57BL/6+others           8     P-GSE50927-3    ArrayExpress
## sample06 Mixed C57BL/6+others           8 Series GSE156692    ArrayExpress
##                   Instrument   ends library_selection library_strategy
## sample01  Illumina HiSeq1500 SINGLE              cDNA          RNA-Seq
## sample02  Illumina HiSeq1500 SINGLE              cDNA          RNA-Seq
## sample03  Illumina HiSeq1500 SINGLE              cDNA          RNA-Seq
## sample04  Illumina HiSeq2000 SINGLE              cDNA          RNA-Seq
## sample05  Illumina HiSeq2000 SINGLE              cDNA          RNA-Seq
## sample06 Illumina NextSeq500 PAIRED              cDNA          RNA-Seq
##          Material_Type  Technology.Type         run
```

```
## sample01     total RNA sequencing assay  ERR3721857
## sample02     total RNA sequencing assay  ERR3721858
## sample03     total RNA sequencing assay  ERR3721859
## sample04     total RNA sequencing assay   SRR988118
## sample05     total RNA sequencing assay   SRR988119
## sample06     total RNA sequencing assay SRR12502790
##                                                                                                  url
## sample01     ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR372/007/ERR3721857/ERR3721857.fastq.gz
## sample02     ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR372/008/ERR3721858/ERR3721858.fastq.gz
## sample03     ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR372/009/ERR3721859/ERR3721859.fastq.gz
## sample04           ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR988/SRR988118/SRR988118.fastq.gz
## sample05           ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR988/SRR988119/SRR988119.fastq.gz
## sample06 ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR125/090/SRR12502790/SRR12502790_1.fastq.gz
##                                                                                                 url2
## sample01
## sample02
## sample03
## sample04
## sample05
## sample06  ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR125/090/SRR12502790/SRR12502790_2.fastq.gz
```

The next part is optional, if you want to change the name of some rows, fix some greek letter (library greekLetters) or modify the name of some group, we can do it as follows:

```
coldata$cell_type[26:36]
```

```
##  [1] "M?B2 "         "M?B2 "         "M?B2 "         "M?B1 "         "M?B1 "
##  [6] "M?B1 "         "M?A_SinglecF" "M?A_SinglecF" "M?A_SinglecF" "M?A_CD11bneg"
## [11] "M?A_CD11bneg"
```

```
for (i in 26:36){
  if(i <= 28){
    coldata$cell_type[i] <- paste("M", greeks("phi"), "B2", sep = "")
    }
  else if(i > 28 & i <= 31){
    coldata$cell_type[i] <- paste("M", greeks("phi"), "B1", sep = "")
  }
  else if(i > 31 & i <= 34){
    coldata$cell_type[i] <- paste("M", greeks("phi"), "A_SinglecF", sep = "")
  }
  else{ coldata$cell_type[i] <- paste("M", greeks("phi"), "A_CD11bneg", sep = "")

  }
}

coldata$cell_type[26:36]
```

```
##  [1] "MfB2"          "MfB2"          "MfB2"          "MfB1"          "MfB1"
##  [6] "MfB1"          "MfA_SinglecF" "MfA_SinglecF" "MfA_SinglecF" "MfA_CD11bneg"
## [11] "MfA_CD11bneg"
```

Next we build a path to any sub folder inside Quants, the subfolder names are in the column “run” from our metadata file, we simply use paste to bind the run name _quant, next file.path will build the path for every file:

```
files <- file.path(dir_files, paste(coldata$run, "_quant", sep=""), "quant.sf")
head(files)
```

```
## [1] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/ERR3721857_quant/quant.sf"
## [2] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/ERR3721858_quant/quant.sf"
## [3] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/ERR3721859_quant/quant.sf"
## [4] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/SRR988118_quant/quant.sf"
## [5] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/SRR988119_quant/quant.sf"
## [6] "C:/Users/lupit/OneDrive/Mouse_sorted_lung_cells/Quants/SRR12502790_quant/quant.sf"
```

The file names can be changed to match the rownames from our metadata file and then we can check if all files exist.

```
names(files) <- rownames(coldata)
all(file.exists(files))
```

```
## [1] TRUE
```

With AnnotationDbi we can build an annotation file to match the gene names given by ensembl.

```
db_object <- read.csv("tx2gene.csv", colClasses=c("NULL",NA,NA)) #skip first column which is an index
head(db_object)
```

```
##                   TXNAME GENENAME
## 1  ENSMUST00000000001.4    Gnai3
## 2 ENSMUST00000000003.13     Pbsn
## 3  ENSMUST00000000010.8    Hoxb9
## 4 ENSMUST00000000028.13    Cdc45
## 5 ENSMUST00000000033.11     Igf2
## 6  ENSMUST00000000049.5     Apoh
```

Once we have everythin ready, we import the files with tximport

```
txi.tx <- tximport(files,
                   type = "salmon",
                   txOut=TRUE)
txi.sum <- summarizeToGene(txi.tx,
                           tx2gene = db_object,
                           countsFromAbundance = c("no", "scaledTPM", "lengthScaledTPM"))
```

```
txi.sum$counts[1000:1005,1:5]
```

```
##       sample01 sample02 sample03 sample04 sample05
## Acsl6   16.000   15.000    5.000   13.000   20.000
## Acsm1  610.000  573.000  724.000  122.000  144.000
## Acsm2    8.186    4.051    6.026    0.000    0.000
## Acsm3   11.000    2.000    4.000   22.094   31.667
## Acsm4    0.000    0.000    0.000    0.000    0.000
## Acsm5    8.196    0.000    0.000    9.000   11.000
```

Before building our DESeq dataset, we can check:

```
all(rownames(coldata) %in% colnames(txi.sum$counts))
```

```
## [1] TRUE
```

If everything is correct, we continue:

```
se <- DESeqDataSetFromTximport(txi = txi.sum,
                               colData = coldata,
                               design = ~ cell_type)
counts(se)[1000:1005,1:5]
```

```
##       sample01 sample02 sample03 sample04 sample05
```

```
## Acsl6       16       15        5       13       20
## Acsm1      610      573      724      122      144
## Acsm2        8        4        6        0        0
## Acsm3       11        2        4       22       32
## Acsm4        0        0        0        0        0
## Acsm5        8        0        0        9       11
```

We can check the object created, looking its dimensions, the rows and columns.

```
dim(se)
```

```
## [1] 29541    55
```

```
head(rownames(se))
```

```
## [1] "0610007P14Rik" "0610009B22Rik" "0610009O20Rik" "0610010F05Rik"
## [5] "0610010K14Rik" "0610011F06Rik"
```

```
head(colData(se))[,1:6]
```

```
## DataFrame with 6 rows and 6 columns
##          project_name    ENA_SAMPLE     batch Sample_source_name  cell_type
##           <character>   <character> <integer>        <character>   <factor>
## sample01  E-MTAB-8573  SAMEA6377060         1            WT lung Whole_lung
## sample02  E-MTAB-8573 SAMEA6377061          1            WT lung Whole_lung
## sample03  E-MTAB-8573  SAMEA6377062         1            WT lung Whole_lung
## sample04 E-GEOD-50927  SAMN02358107         2            WT lung Whole_lung
## sample05 E-GEOD-50927  SAMN02358107         2            WT lung Whole_lung
## sample06    GSE156692  SAMN15888043         3            WT lung Whole_lung
##          sample_description
##                 <character>
## sample01 lung cell unsorted
## sample02 lung cell unsorted
## sample03 lung cell unsorted
## sample04 lung cell unsorted
## sample05 lung cell unsorted
## sample06 lung cell unsorted
```

```
summary(se$cell_type)
```

```
##  Endothelial   Epithelial   Fibroblast   Leukocytes  Mesothelial MfA_CD11bneg
##            5            8            3            5            3            2
## MfA_SinglecF         MfB1         MfB2     Monocyte  Neutrophils      Stromal
##            3            3            3            2            3            5
##   Whole_lung
##           10
```

Exploratory analysis and visualization

```
nrow(se)
```

```
## [1] 29541
```

```
keep <- rowSums(counts(se)) > 1
se <- se[keep,]
nrow(se)
```

```
## [1] 24224
```

Then we normalize the counts, the normalization factors matrix should not have 0's in it # it should have

geometric mean near 1 for each row

```
normFactors <- matrix(runif(nrow(se)*ncol(se),0.5,1.5),
                      ncol=ncol(se),nrow=nrow(se),
                      dimnames=list(1:nrow(se),1:ncol(se)))
normFactors <- normFactors / exp(rowMeans(log(normFactors)))
se <- DESeq(se)
se <- estimateSizeFactors(se)
```

Since we are using different batches of RNAseq experiments, is recommendable to reduce the covariative effect of it. The tool ComBat_seq from sva library help us with this issue. More info: https://rdrr.io/bioc/sva/man/ComBat.html

```
count_matrix <- counts(se, normalized=TRUE)
batch <- se$batch

adjusted <- sva::ComBat_seq(count_matrix, batch, NULL)
```

```
## Found 7 batches
## Using null model in ComBat-seq.
## Adjusting for 0 covariate(s) or covariate level(s)
## Estimating dispersions
## Fitting the GLM model
## Shrinkage off - using GLM estimates for parameters
## Adjusting the data
```

```
adjusted <- as.data.frame(lapply(as.data.frame(adjusted), as.integer))  #normalization cause non intege
genes <- rownames(count_matrix)
adjusted <- cbind(genes,adjusted)
se <- DESeqDataSetFromMatrix(countData=adjusted,
                             colData=coldata,
                             design=~cell_type, tidy = TRUE)
```

Next we migth want to use a control group or reference, first we check the current levels:

```
levels(se$cell_type)
```

```
##  [1] "Endothelial"  "Epithelial"   "Fibroblast"   "Leukocytes"   "Mesothelial"
##  [6] "MfA_CD11bneg" "MfA_SinglecF" "MfB1"         "MfB2"         "Monocyte"
## [11] "Neutrophils"  "Stromal"      "Whole_lung"
```

I want to use Whole_lung as reference, so we use magrittr relevel:

```
se$cell_type %<>% relevel("Whole_lung")
levels(se$cell_type)
```

```
##  [1] "Whole_lung"   "Endothelial"  "Epithelial"   "Fibroblast"   "Leukocytes"
##  [6] "Mesothelial"  "MfA_CD11bneg" "MfA_SinglecF" "MfB1"         "MfB2"
## [11] "Monocyte"     "Neutrophils"  "Stromal"
```

At this point we can save our counts table to use it later or with online tools to visualize it. Uncomment to run it.

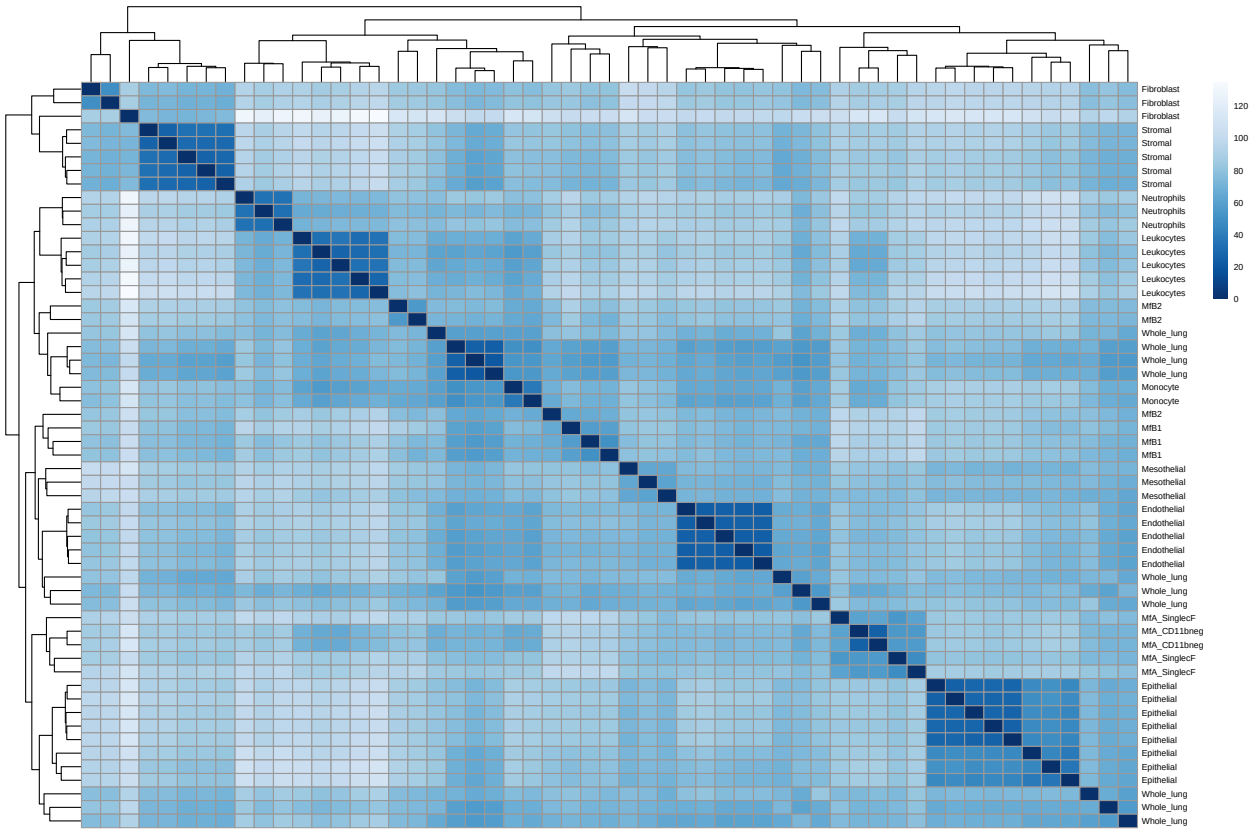
```
#write.csv(counts(se), "gene_counts_norm_batchfix.csv", quote = FALSE)
```

We can evaluate the similarity between every sample and the plot it with a heatmap. First we must transform the data by calling vst function from DESeq2. More info: https://www.rdocumentation.org/packages/DESeq2/versions/1.12.3/topics/vst Then, we calculate the similiraty distance, the default is Euclidian distance but we can try other formulas.

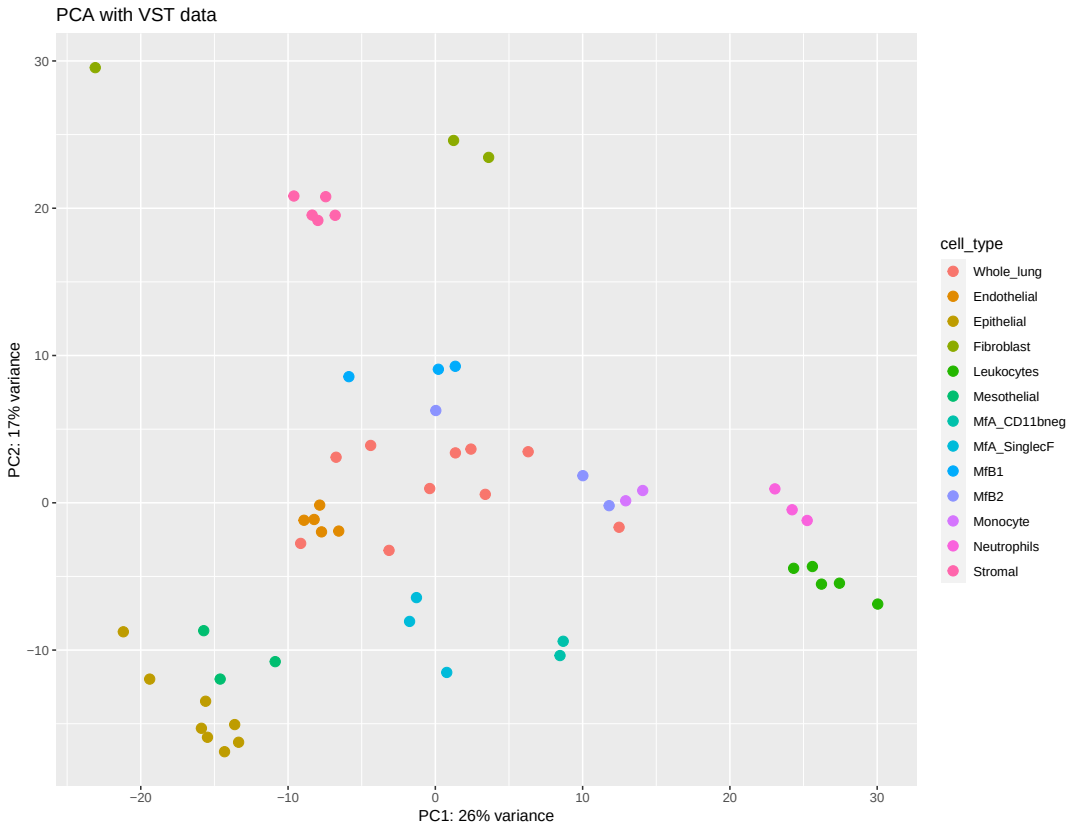
```
#Data transformation
vsd <- vst(se, blind = FALSE)
#Asses overall similarity between samples
sample_dist <- dist(t(assay(vsd)))
```

Visualize the result: Heatmap of sample-to-sample distances using the variance stabilizing transformed values

```
sampleDistMatrix <- as.matrix(sample_dist)
rownames(sampleDistMatrix) <- paste( vsd$cell_type, sep = " - " )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows = sample_dist,
         clustering_distance_cols = sample_dist,
         col = colors,
         fontsize = 6)
```




```
pcaData <- plotPCA(vsd, intgroup = c("cell_type"), returnData = TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(x = PC1, y = PC2, color = cell_type)) +
  geom_point(size =3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA with VST data")
```



Diferential expression analysis

```
dds <- DESeq(se)
res <- results(dds)
head(res)
```

```
## log2 fold change (MLE): cell type Stromal vs Whole lung
## Wald test p-value: cell type Stromal vs Whole lung
## DataFrame with 6 rows and 6 columns
##                baseMean log2FoldChange     lfcSE       stat     pvalue
##               <numeric>      <numeric> <numeric>  <numeric>  <numeric>
## 0610007P14Rik   484.336    -0.11527890  0.177486 -0.6495109 0.51600820
## 0610009B22Rik   158.898     0.14297173  0.283996  0.5034281 0.61466331
## 0610009O20Rik   667.770     0.00577567  0.140396  0.0411385 0.96718550
## 0610010F05Rik   369.120     0.48534955  0.183713  2.6418859 0.00824458
## 0610010K14Rik   561.971    -0.34138176  0.145505 -2.3461825 0.01896682
## 0610011F06Rik   372.716    -0.20787520  0.262197 -0.7928198 0.42788283
##                     padj
##                <numeric>
## 0610007P14Rik 0.6404735
## 0610009B22Rik 0.7237376
## 0610009O20Rik 0.9784500
## 0610010F05Rik 0.0308087
## 0610010K14Rik 0.0576909
## 0610011F06Rik 0.5608892
```

We subset the results table to these genes and then sort it by the log2 fold change

```
res_significant <- subset(res, padj < 0.1)
```

```
head(res_significant[order(res_significant$log2FoldChange), ])
```

```
## log2 fold change (MLE): cell type Stromal vs Whole lung
## Wald test p-value: cell type Stromal vs Whole lung
## DataFrame with 6 rows and 6 columns
##          baseMean log2FoldChange     lfcSE      stat      pvalue        padj
##         <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
## Gm10184  85.44453       -29.9987   3.27635  -9.15615 5.37829e-20 2.33661e-17
## Cyp26b1  15.07910       -29.8153   5.93083  -5.02717 4.97778e-07 1.33967e-05
## Phf20l1  39.49918       -29.7571   5.93068  -5.01747 5.23551e-07 1.38855e-05
## Ddx41    36.96043       -29.6902   5.93079  -5.00612 5.55393e-07 1.46027e-05
## Zfp458   21.56144       -29.3169   4.49361  -6.52413 6.83955e-11 7.16807e-09
## Gm10715   7.83707       -29.2735   3.86500  -7.57400 3.61906e-14 7.76906e-12
```

```
head(res_significant[order(res_significant$log2FoldChange, decreasing = TRUE), ])
```

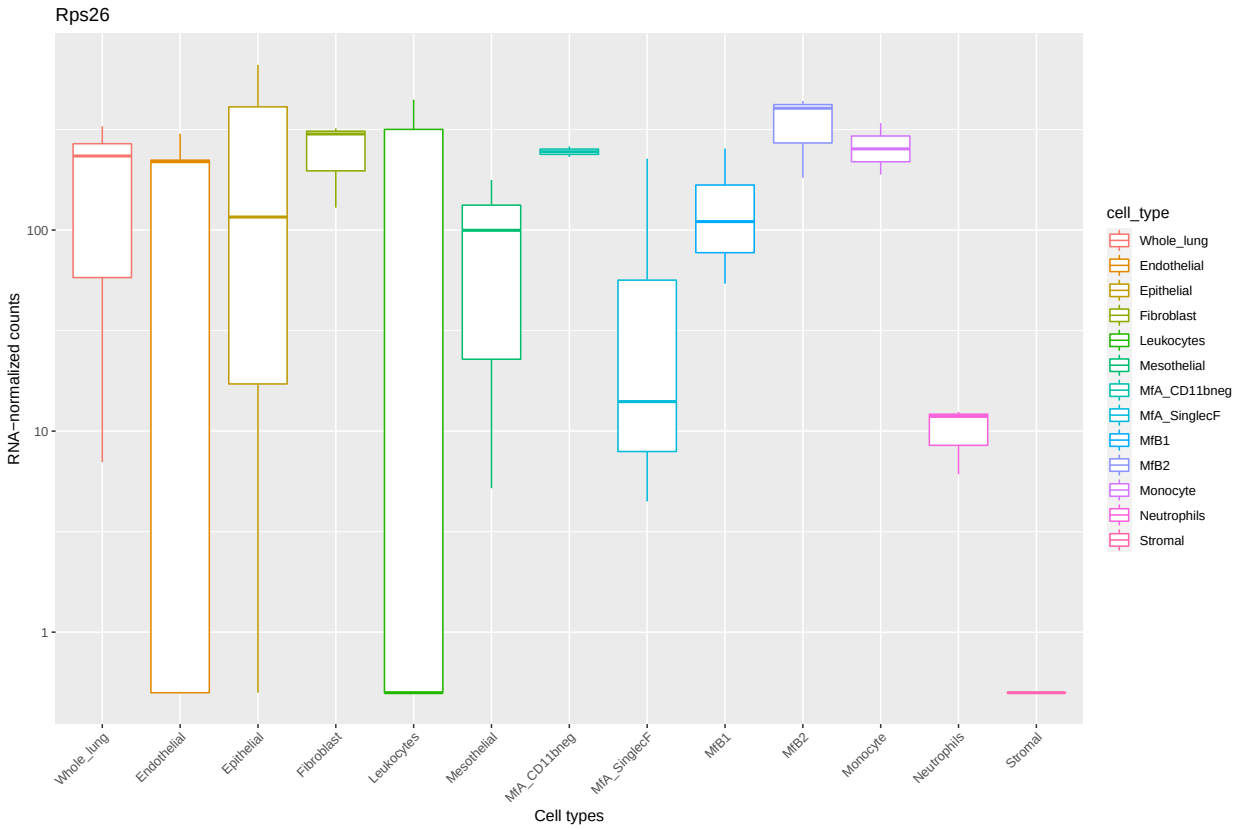
```
## log2 fold change (MLE): cell type Stromal vs Whole lung
## Wald test p-value: cell type Stromal vs Whole lung
## DataFrame with 6 rows and 6 columns
##          baseMean log2FoldChange     lfcSE      stat      pvalue        padj
##         <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
## Zfp821    31.5945       20.69252  3.715806   5.56878 2.56524e-08 1.19490e-06
## Sult1e1   54.8266       10.87601  1.500720   7.24720 4.25488e-13 7.61165e-11
## Ddx3y   1014.0233       10.36890  0.805753  12.86858 6.76373e-38 1.76311e-34
## Nkx6-1    26.5119        7.68176  1.047437   7.33386 2.23613e-13 4.34071e-11
## Creb5    106.9313        7.09551  1.255400   5.65199 1.58601e-08 8.17511e-07
## Pth1r     37.1080        6.39440  2.256907   2.83326 4.60760e-03 1.99561e-02
```

Plotting the results

```
myplot <- function(data, title){
  ggplot(data, aes(x = cell_type, y = count, color = cell_type)) +
    scale_y_log10() +
    geom_boxplot() +
    labs(title=title) +
    ylab("RNA-normalized counts") +
    xlab("Cell types") +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
  }
```

```
topGene <- rownames(res)[which.min(res$padj)]
topGene_count <- plotCounts(dds, gene = topGene, intgroup=c("cell_type"), returnData = TRUE)
```

```
myplot(topGene_count, topGene)
```



Ism1 and genes of proteins coexpressed by flowcytometry

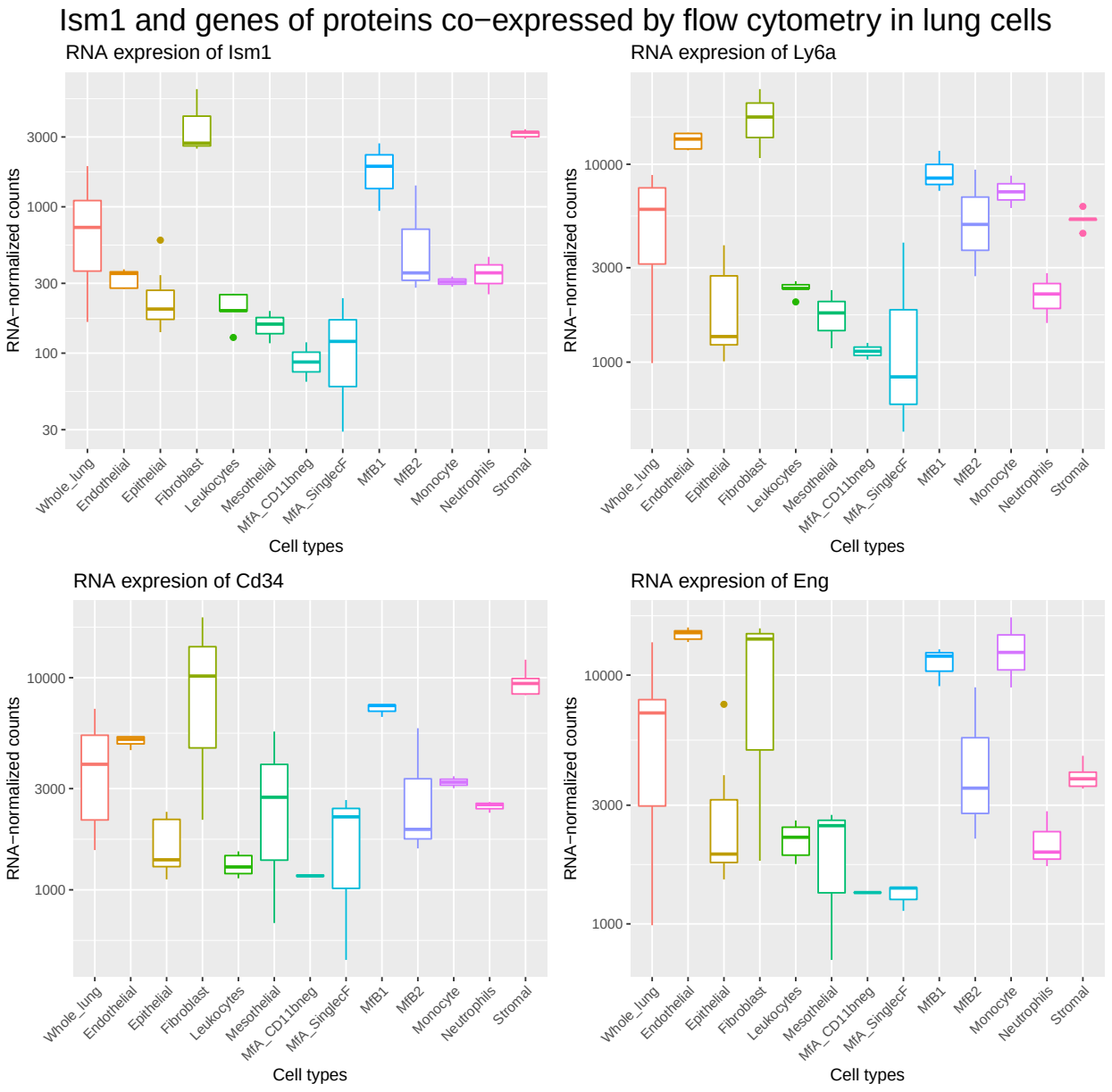
We can count and plot any gene we're interested in, in this case we looked for Isthmin-1 and some co-expressed protein genes like Sca1(Ly6a) or CD105(Eng).

```
multi_count <- function(data, gen_name){
  plotCounts(data,
             gene = gen_name,
             intgroup=c("cell_type"),
             returnData = TRUE)
}
```

```
gen_name <- c("Ism1", "Ly6a", "Cd34", "Eng")

# counting multiple genes from gen_name
list_c <- vector('list', length(gen_name))
for (i in seq_along(gen_name)){
  list_c[[i]] <- multi_count(dds, gen_name[i])
}

#multiple plots from list_c
list_p <- vector('list', length(gen_name))

for (i in seq_along(list_c)){
  list_p[[i]] <- myplot(list_c[[i]], paste0("RNA expresion of ", gen_name[i]))
  list_p[[i]] <- list_p[[i]] + theme(legend.position = "none") #suppr legends
}
```

After building the list of gene counts and gene plots, we use gridExtra to arrange them into one plot as follows:

```
title_exp <- "Ism1 and genes of proteins co-expressed by flow cytometry in lung cells"
plotSW <- gridExtra::grid.arrange(grobs = list_p,
                                  top = textGrob(title_exp, gp =gpar(fontsize=20)),
                                  ncol=2,
                                  bottom = "Ly6a(Sca1), Eng(CD105)")
```

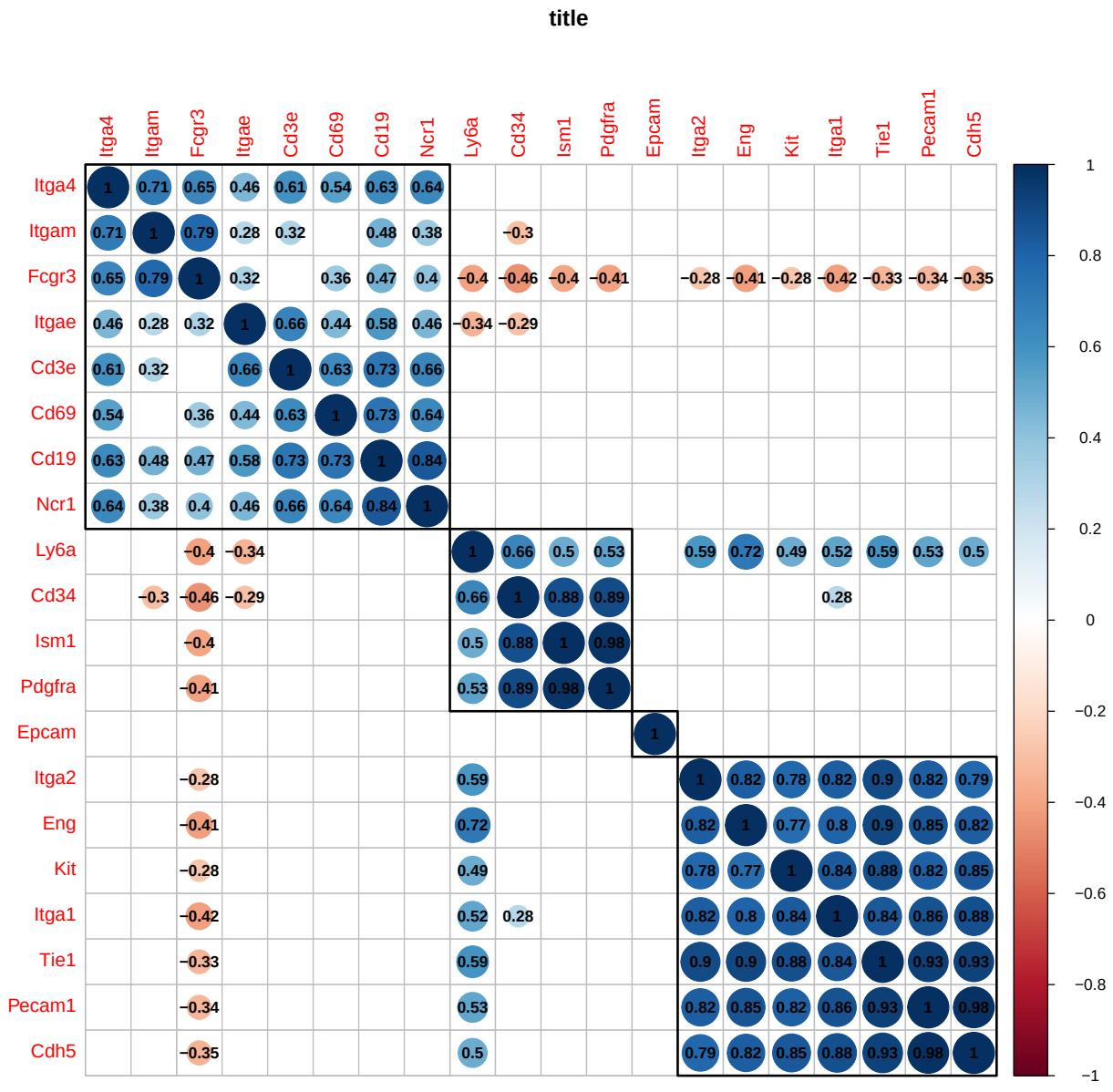


We can seek for gene correlations across the results, the following list corresponde to genes assesed by flow cytometry to identify hematopoietic subsets, mesenchymal stem cells, endothelial cells and, epithelial cells.

```
genes <- c("Ism1", "Itga1", "Itga2", "Itga4","Itgae", "Itgam", "Cd3e", "Cd19", "Cd69", "Fcgr3", "Ncr1",
```

```
#Count all the dds results
nc <- counts(dds,normalized=TRUE)
#Transpose the matrix to have genxgen
nc_t <- t(nc)
```

```
## leave blank on non-significant coefficient
## add significant correlation coefficients
M <- cor(nc_t[, genes])
testRes = cor.mtest(nc_t[, genes], conf.level = 0.95)
corrplot(M, p.mat = testRes$p, method = 'circle', insig='blank',
         addCoef.col ='black', title="title", mar=c(0,0,1,0), number.cex = 0.8, order = 'hclust', addre
```



#Ism1 and associated genes to Fibroblasts

Since we saw Ism1 gene expressin was high in Fibroblasts, we wanted to see typical flow cytometry marker for this subset. Flow cytometry experiments also had showed Ism1 co-expression with CD34 and Sca1, antigens often found in fibroblasts

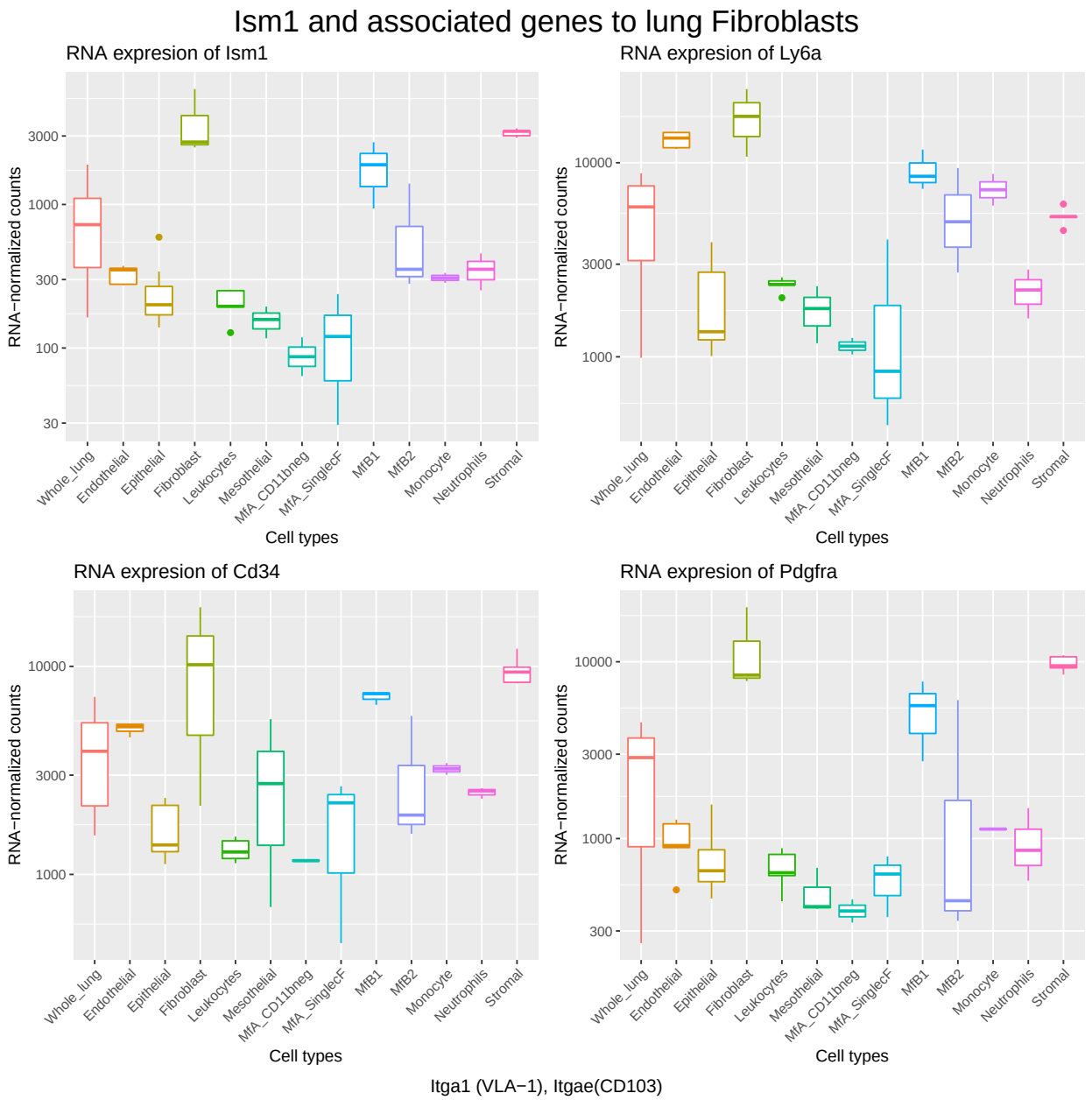
```
plotCounts(dds,
             gene = "Ly6a",
             intgroup=c("cell_type"),
             returnData = T)
```

```
##               count    cell_type
## sample01   988.7010   Whole_lung
## sample02  8176.8747   Whole_lung
## sample03  6811.8215   Whole_lung
## sample04  1852.2242   Whole_lung
## sample05  8826.1885   Whole_lung
## sample06  2661.3008   Whole_lung
## sample07  6627.4962   Whole_lung
## sample08  7876.4945   Whole_lung
## sample09  5105.5344   Whole_lung
## sample10  5286.6543   Whole_lung
## sample11 11765.6107  Endothelial
## sample12 14431.3265  Endothelial
## sample13 14308.9680  Endothelial
## sample14 13405.4958  Endothelial
## sample15 11949.9423  Endothelial
## sample16  2564.8760   Leukocytes
## sample17  2350.1212   Leukocytes
## sample18  2357.6790   Leukocytes
## sample19  2014.8328   Leukocytes
## sample20  2460.5193   Leukocytes
## sample21  2210.0902  Neutrophils
## sample22  1577.0121  Neutrophils
## sample23  2815.5329  Neutrophils
## sample24  6010.3704     Monocyte
## sample25  8757.3157     Monocyte
## sample26  9387.8676         MfB2
## sample27  2717.6553         MfB2
## sample28  4964.7218         MfB2
## sample29  8512.9912         MfB1
## sample30  7340.6998         MfB1
## sample31 11693.1225         MfB1
## sample32   842.0350 MfA_SinglecF
## sample33  4018.2474 MfA_SinglecF
## sample34   445.1154 MfA_SinglecF
## sample35  1029.1971 MfA_CD11bneg
## sample36  1251.0820 MfA_CD11bneg
## sample37  1204.4986   Epithelial
## sample38  1229.0118   Epithelial
## sample39  1416.5726   Epithelial
## sample40  1009.5319   Epithelial
## sample41  1279.9605   Epithelial
## sample42  3895.5120   Epithelial
## sample43  2445.4272   Epithelial
## sample44  3790.8350   Epithelial
## sample45 23969.7657   Fibroblast
```

```
## sample46 10748.1449   Fibroblast
## sample47 17329.4275   Fibroblast
## sample48  2311.7259  Mesothelial
## sample49  1772.7422  Mesothelial
## sample50  1174.5519  Mesothelial
## sample51  4476.9035      Stromal
## sample52  6120.9926      Stromal
## sample53  5285.5223      Stromal
## sample54  5222.3533      Stromal
## sample55  5290.6109      Stromal
```

```
gen_name2 <- c("Ism1", "Ly6a", "Cd34", "Pdgfra")

# counting multiple genes from gen_name
list_c <- vector('list', length(gen_name2))
for (i in seq_along(gen_name2)){
  list_c[[i]] <- multi_count(dds, gen_name2[i])
}

#multiple plots from list_c
list_p <- vector('list', length(gen_name2))

for (i in seq_along(list_c)){
  list_p[[i]] <- myplot(list_c[[i]], paste0("RNA expresion of ", gen_name2[i]))
  list_p[[i]] <- list_p[[i]] + theme(legend.position = "none") #suppr legends
}
```

```
#arrange every plot from the list_p
title_exp <-"Ism1 and associated genes to lung Fibroblasts"

plotSW <- gridExtra::grid.arrange(grobs = list_p,
                                  top = textGrob(title_exp, gp =gpar(fontsize=20)),
                                  ncol=2,
                                  bottom = "Itga1 (VLA-1), Itgae(CD103)")
```



We could look for fibroblast differntially expressed genes compared to whole lung cells.

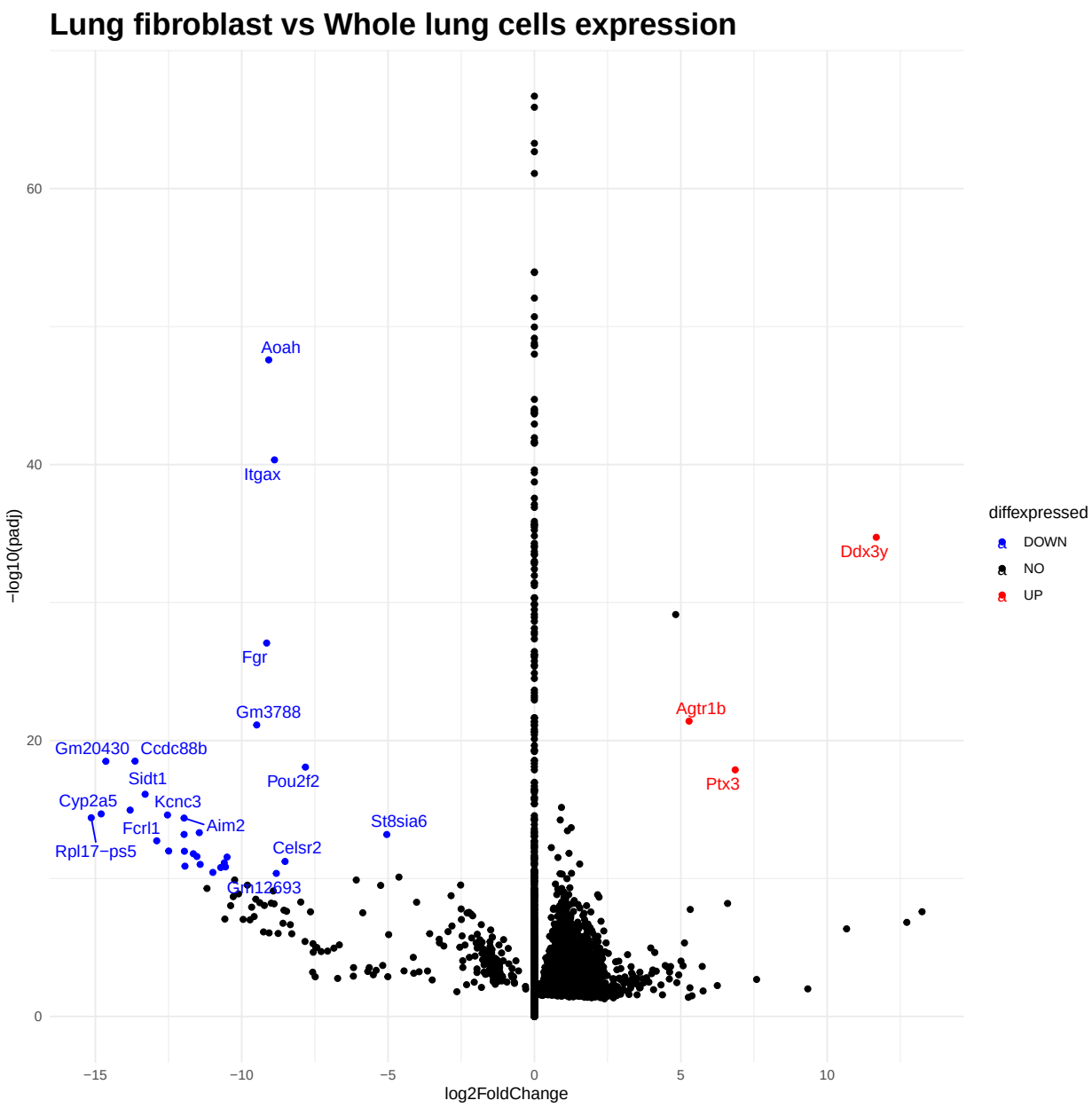
```
res_fibroblast <- lfcShrink(dds,
                            coef="cell_type_Fibroblast_vs_Whole_lung",
                            type="apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

We will keep only those genes 5 times higher/lower with a padj < 10e-5

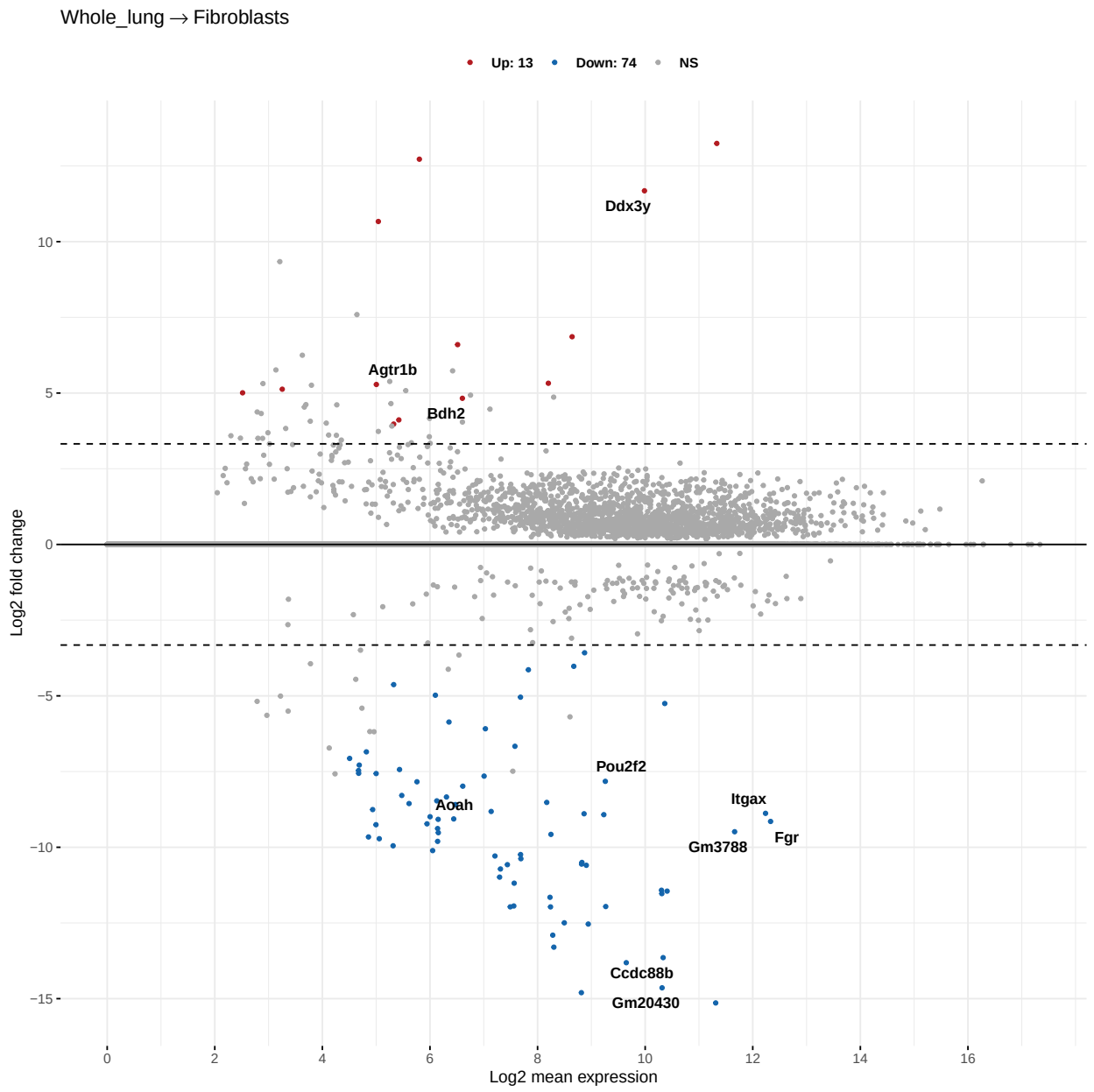
```
# add a column of NAs
res_fibroblast_df <- as.data.frame(res_fibroblast)
res_fibroblast_df$gene_name <- rownames(res_fibroblast)
```

```r
res_fibroblast_df$diffexpressed <- "NO" #Non variable genes

# if log2Foldchange > 5 and pvalue < 0.0001, set as "UP"
res_fibroblast_df$diffexpressed[res_fibroblast_df$log2FoldChange > 5 & -log10(res_fibroblast_df$padj) >

# if log2Foldchange < -5 and pvalue < 0.0001, set as "DOWN"
res_fibroblast_df$diffexpressed[res_fibroblast_df$log2FoldChange < -5 & -log10(res_fibroblast_df$padj) >

# Now write down the name of genes beside the points...
# Create a new column "delabel" to de, that will contain the name of genes differentially expressed (NA
res_fibroblast_df$delabel <- NA
res_fibroblast_df$delabel[res_fibroblast_df$diffexpressed != "NO"] <- res_fibroblast_df$gene_name[res_fi

#plot
#enhance labeling distances
ggplot(res_fibroblast_df, aes(x=log2FoldChange, y=-log10(padj), col=diffexpressed, label=delabel)) +
  geom_point()+
  theme_minimal() +
  scale_color_manual(values = c("blue", "black", "red")) +
  geom_text_repel() +
  ggtitle("Lung fibroblast vs Whole lung cells expression") +
    theme(plot.title = element_text(size = 20, face = "bold"))
```




```
#ggpubr MA plot

# add a column of NAs
res_fibroblast_df <- as.data.frame(res_fibroblast)
res_fibroblast_df$gene_name <- rownames(res_fibroblast)
res_fibroblast_df$detection_call <- 0 #Non variable genes

# if log2 mean exp > 4 considered as expressed
res_fibroblast_df$detection_call[log2(res_fibroblast_df$baseMean) > 4] <- 1

options(ggrepel.max.overlaps = Inf)
# ggpubr::ggmaplot fdr = padj threshold     fc = foldchange threshold
ggmaplot(res_fibroblast, main = expression("Whole_lung" %->% "Fibroblasts"),
         fdr = 0.0001, fc = 10, size = 1,
```

```
        palette = c("#B31B21", "#1465AC", "darkgray"),
        genenames = as.vector(res_fibroblast$gene_name),
        legend = "top", top = 10,
        font.label = c("bold", 11),
        font.legend = "bold",
        font.main = "bold",
        ggtheme = ggplot2::theme_minimal())
```

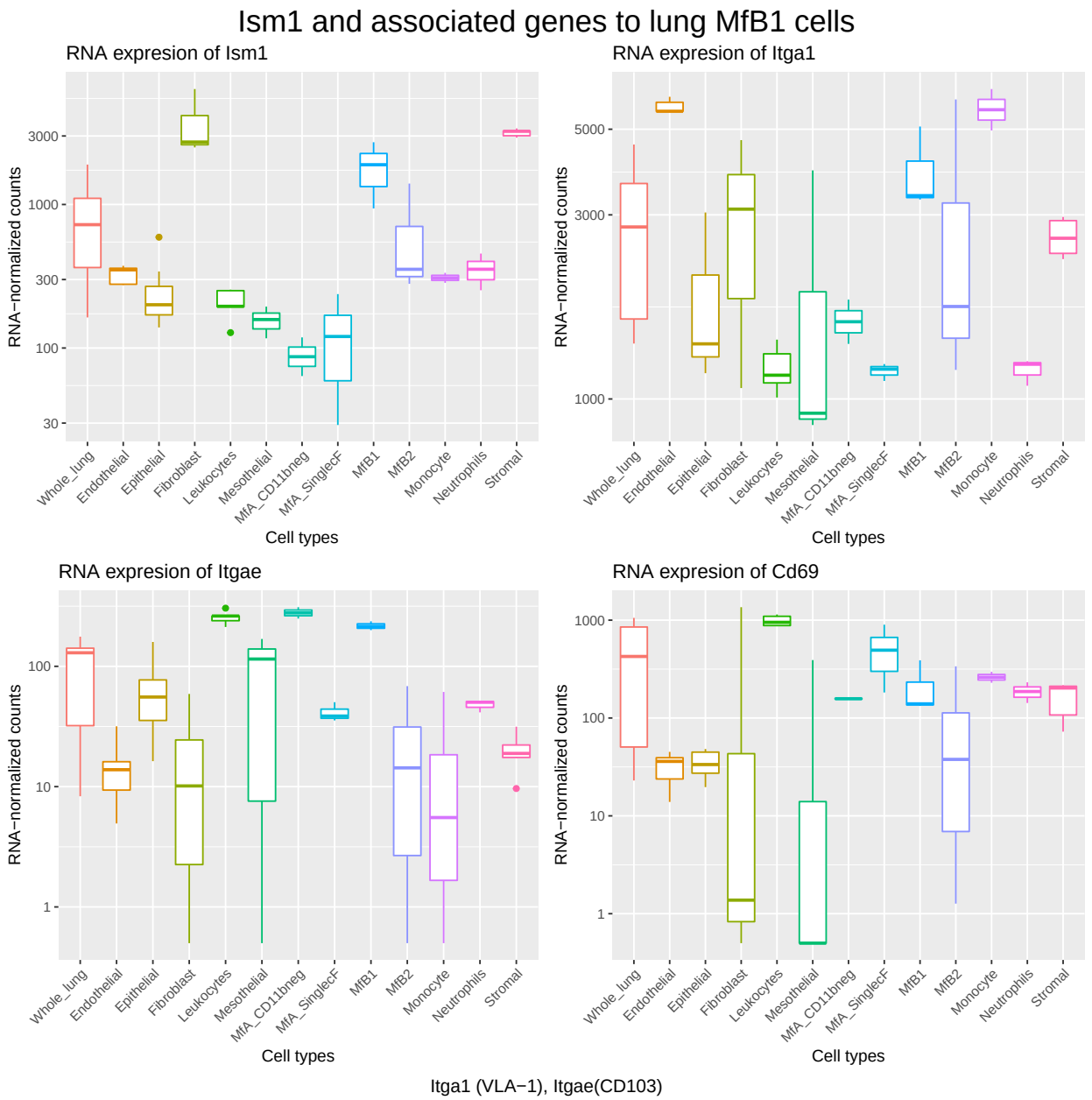


#Ism1 and associated genes to Macrophages B1

Since we saw Ism1 gene expressin was high in Macrophafes B1, we wanted to see typical flow cytometry marker for this subset. Flow cytometry experiments also had showed Ism1 co-expression with CD49a (Itga1), CD103(Itgae) in lungs (Unpublished data). Also CD69 is highly expressed by this subpopulation.

```r
gen_name3 <- c("Ism1", "Itga1", "Itgae", "Cd69")

list_c <- vector('list', length(gen_name3))
for (i in seq_along(gen_name3)){
  list_c[[i]] <- multi_count(dds, gen_name3[i])
}

list_p <- vector('list', length(gen_name3))
for (i in seq_along(list_c)){
  list_p[[i]] <- myplot(list_c[[i]], paste0("RNA expresion of ", gen_name3[i]))
  list_p[[i]] <- list_p[[i]] + theme(legend.position = "none") #suppr legends
}
```

```r
# arrange every plot from the list_p
title_exp <- paste("Ism1 and associated genes to lung M", greeks("phi"), "B1 cells", sep="")

plotSW <- gridExtra::grid.arrange(grobs = list_p,
                                  top = textGrob(title_exp, gp =gpar(fontsize=20)),
                                  ncol=2,
                                  bottom = "Itga1 (VLA-1), Itgae(CD103)")
```



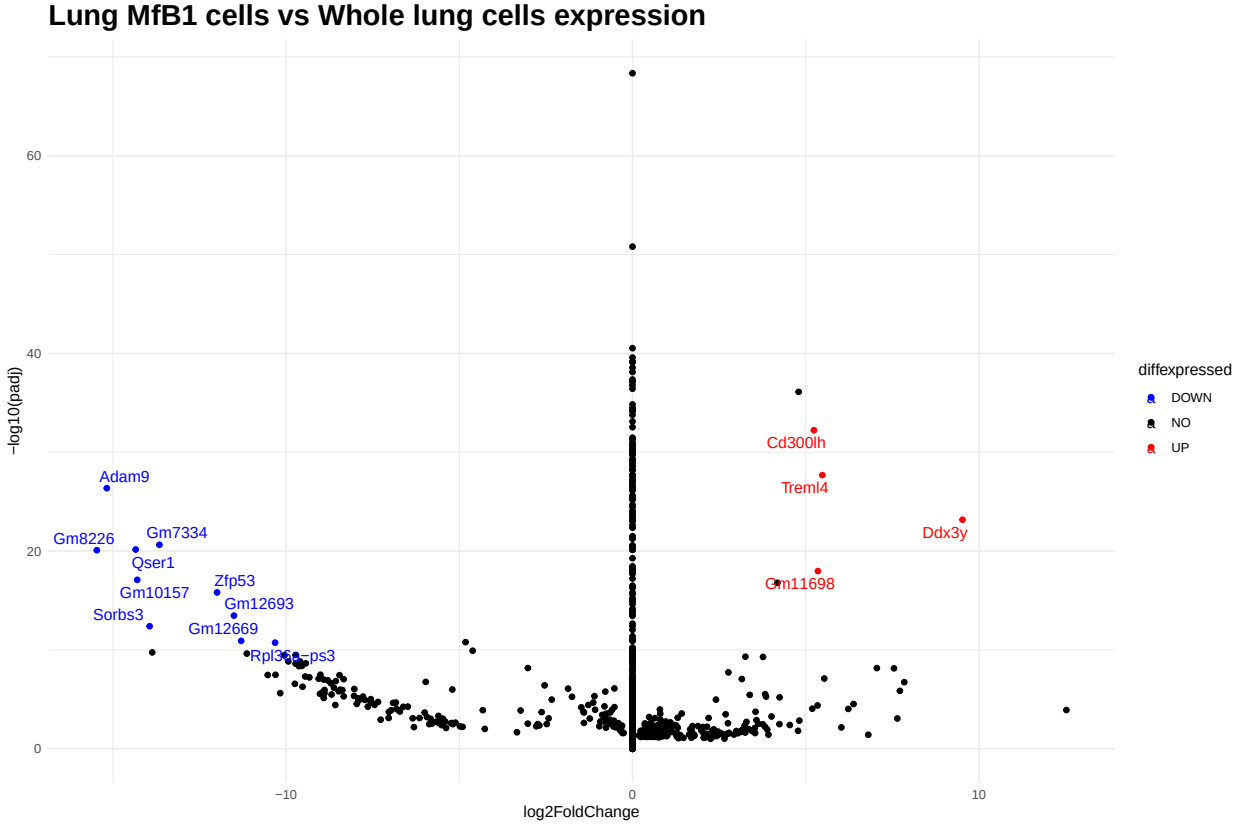
We could look for fibroblast differntially expressed genes compared to whole lung cells.

```
res_MfB1 <- lfcShrink(dds,
                      coef="cell_type_MfB1_vs_Whole_lung",
                      type="apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
# add a column of NAs
res_MfB1_df <- as.data.frame(res_MfB1)
res_MfB1_df$gene_name <- rownames(res_MfB1)
res_MfB1_df$diffexpressed <- "NO" #Non variable genes
```

```r
# if log2Foldchange > 5 and pvalue < 0.0001, set as "UP"
res_MfB1_df$diffexpressed[res_MfB1_df$log2FoldChange > 5 & -log10(res_MfB1_df$padj) > 10] <- "UP"

# if log2Foldchange < -5 and pvalue < 0.0001, set as "DOWN"
res_MfB1_df$diffexpressed[res_MfB1_df$log2FoldChange < -5 & -log10(res_MfB1_df$padj) > 10] <- "DOWN"

# Now write down the name of genes beside the points...
# Create a new column "delabel" to de, that will contain the name of genes differentially expressed (NA
res_MfB1_df$delabel <- NA
res_MfB1_df$delabel[res_MfB1_df$diffexpressed != "NO"] <- res_MfB1_df$gene_name[res_MfB1_df$diffexpresse

#plot
#enhance labeling distances
ggplot(res_MfB1_df, aes(x=log2FoldChange, y=-log10(padj), col=diffexpressed, label=delabel)) +
  geom_point()+
  theme_minimal() +
  scale_color_manual(values = c("blue", "black", "red")) +
  geom_text_repel() +
  ggtitle(paste0(paste0("Lung M", greeks("phi"), "B1 cells vs Whole lung cells expression"))) +
    theme(plot.title = element_text(size = 20, face = "bold"))
```




```r
# add a column of NAs
res_MfB1_df <- as.data.frame(res_MfB1)
res_MfB1_df$gene_name <- rownames(res_MfB1)
res_MfB1_df$detection_call <- 0 #Non variable genes
```

```r
# if log2 mean exp > 4 considered as expressed
res_MfB1_df$detection_call[log2(res_MfB1_df$baseMean) > 4] <- 1

options(ggrepel.max.overlaps = Inf)
# ggpubr::ggmaplot fdr = padj threshold     fc = foldchange threshold
ggmaplot(res_MfB1_df, main = expression("Whole_lung" %->% paste("M", phi , "B1 cells")),
         fdr = 0.0001, fc = 10, size = 1,
         palette = c("#B31B21", "#1465AC", "darkgray"),
         genenames = as.vector(res_MfB1_df$gene_name),
         legend = "top", top = 10,
         font.label = c("bold", 11),
         font.legend = "bold",
         font.main = "bold",
         ggtheme = ggplot2::theme_minimal())
```