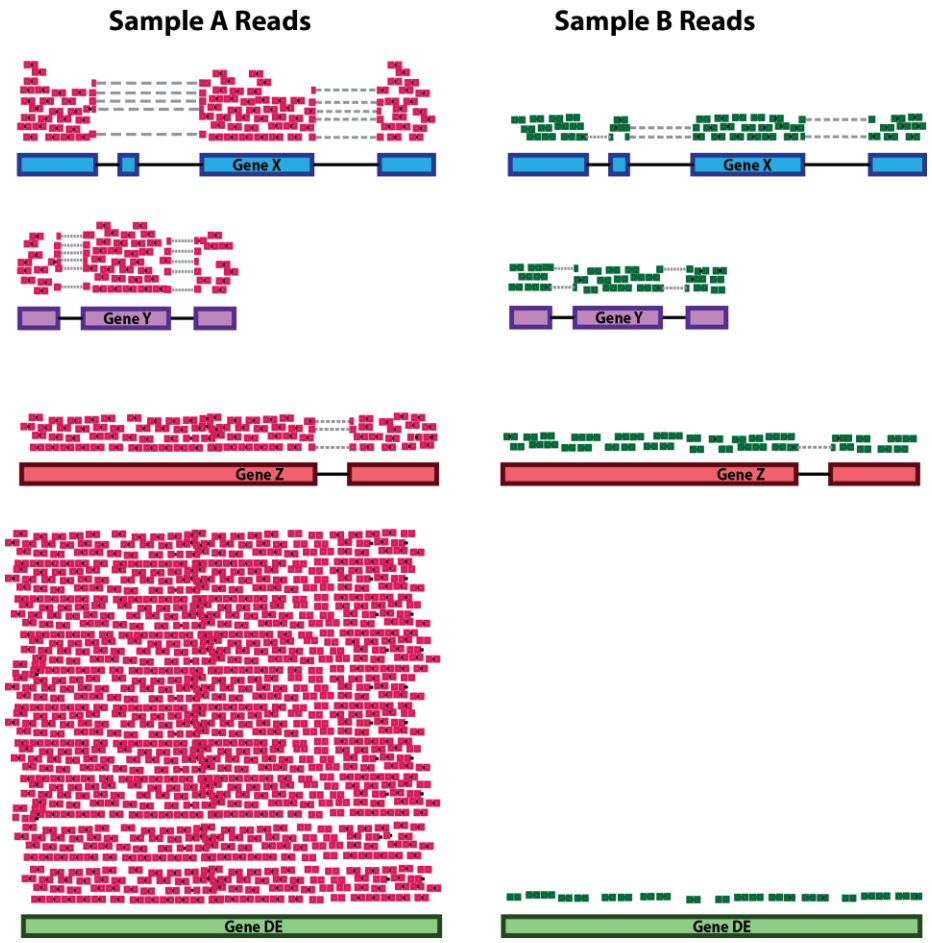
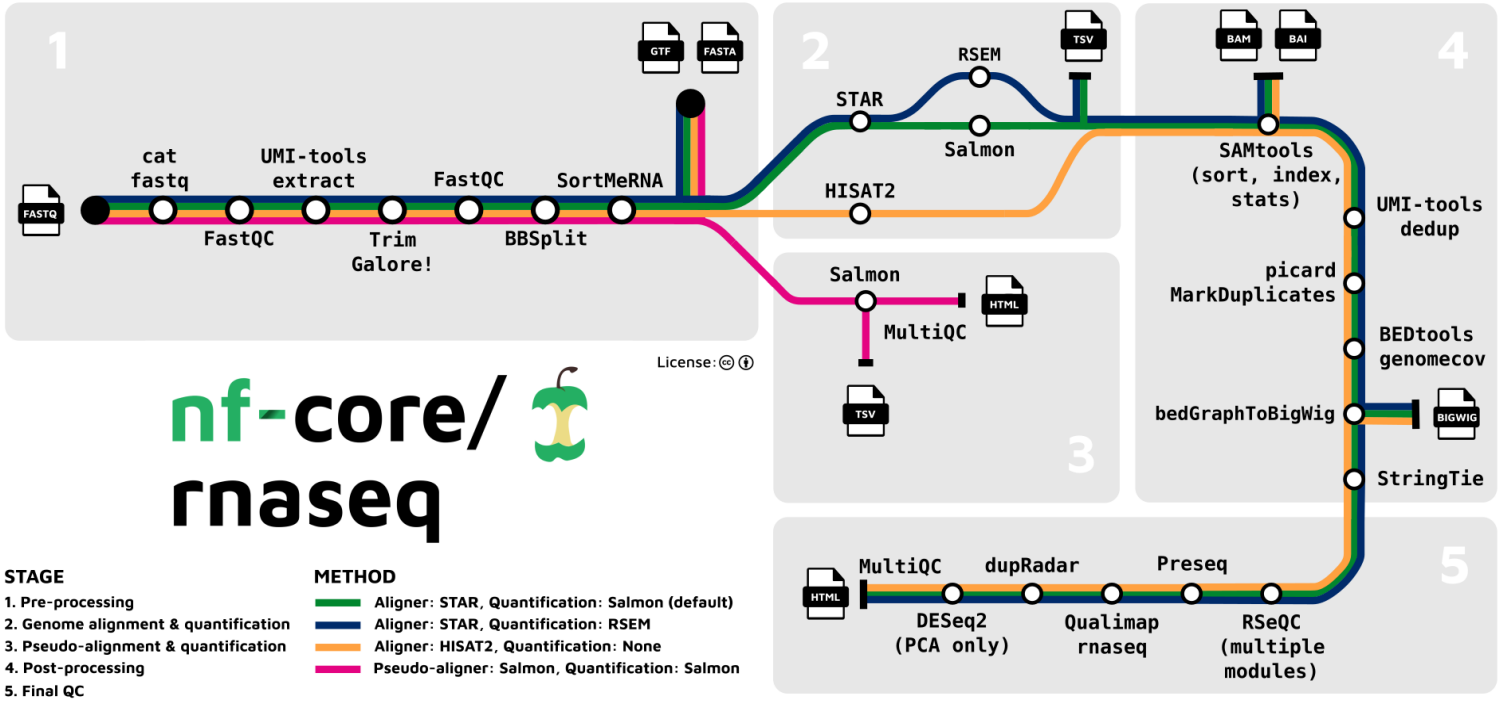
RNA-Seq
Differential Expression Analysis

Myles Lewis
Professor of Precision Medicine & Rheumatology

Cankut Cubuk
Senior Bioinformatician

RNA-seq


Sample A Reads
Sample B Reads
Gene X
Gene Y
Gene Z
Gene DE


1
2
3
4
5
GTF
FASTA
FASTQ
cat fastq
FastQC
UMI-tools extract
Trim Galore!
FastQC
SortMeRNA
BBSplit
STAR
RSEM
Salmon
TSV
HISAT2
Salmon
MultiQC
HTML
TSV
BAM
BAI
SAMtools (sort, index, stats)
UMI-tools dedup
picard MarkDuplicates
BEDtools genomecov
bedGraphToBigWig
BIGWIG
StringTie
HTML
MultiQC
dupRadar
Preseq
DESeq2 (PCA only)
Qualimap rnaseq
RSeQC (multiple modules)
License:
nf-core/rnaseq
STAGE
1. Pre-processing
2. Genome alignment & quantification
3. Pseudo-alignment & quantification
4. Post-processing
5. Final QC
METHOD
Aligner: STAR, Quantification: Salmon (default)
Aligner: STAR, Quantification: RSEM
Aligner: HISAT2, Quantification: None
Pseudo-aligner: Salmon, Quantification: Salmon

Group comparisons

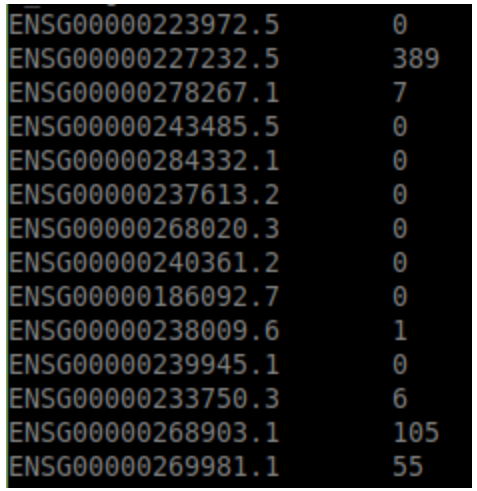
- Specialist tools depending on data type
- Microarray = limma
- **RNA-Seq** = DESeq2, edgeR, limma voom
- Single cell RNA-Seq = Seurat

General concept

- Use raw read counts.
- Prepare a design matrix or formula to specify the groups to be compared.
- Perform DEG analysis.
- Functional annotation of genes
- Visualization of the results

Read counts

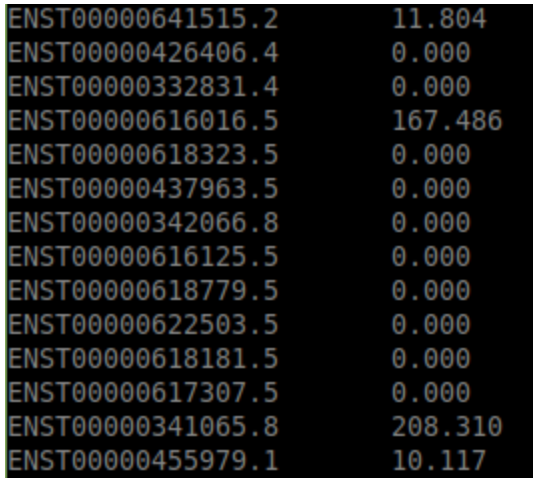
- STAR, Salmon, Kalisto,HISAT2



```
ENSG00000223972.5       0
ENSG00000227232.5       389
ENSG00000278267.1       7
ENSG00000243485.5       0
ENSG00000284332.1       0
ENSG00000237613.2       0
ENSG00000268020.3       0
ENSG00000240361.2       0
ENSG00000186092.7       0
ENSG00000238009.6       1
ENSG00000239945.1       0
ENSG00000233750.3       6
ENSG00000268903.1       105
ENSG00000269981.1       55
```

Merge files manually

```
dir <- "/home/username/Downloads/"

myOutTabs <- list.files(dir, pattern = "ReadsPerGene.tab", full.names = T)
gexDF <- list()

for(f in myOutTabs){
  myGex <- read.delim(f, header = F)
  gexDF[[basename(f)]]  <- myGex$V2
}

cts <- do.call("cbind",gexDF)
```

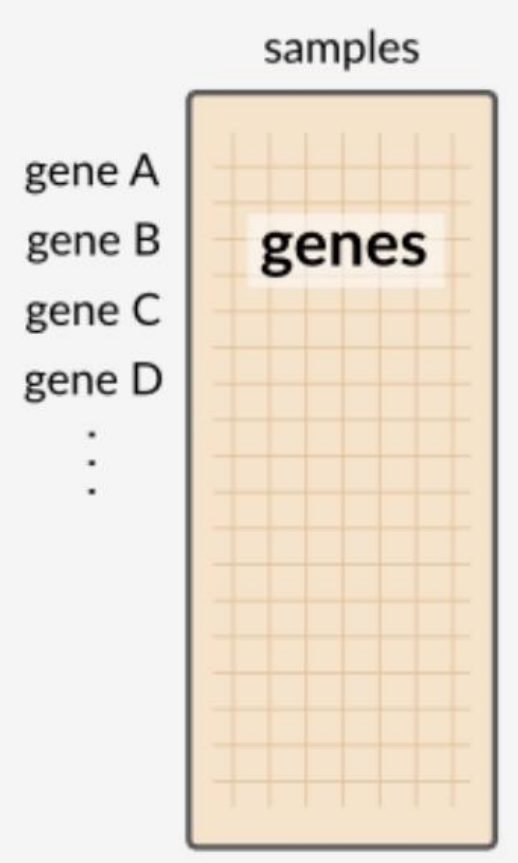


```
ENST00000641515.2       11.804
ENST00000426406.4       0.000
ENST00000332831.4       0.000
ENST00000616016.5       167.486
ENST00000618323.5       0.000
ENST00000437963.5       0.000
ENST00000342066.8       0.000
ENST00000616125.5       0.000
ENST00000618779.5       0.000
ENST00000622503.5       0.000
ENST00000618181.5       0.000
ENST00000617307.5       0.000
ENST00000341065.8       208.310
ENST00000455979.1       10.117
```

Import transcript levels data using R packages

```
library(tximport)

dir <- "/home/username/Downloads/"
files <- file.path(dir, "salmon", samples, "quant.sf")
txi <- tximport(files, type = "salmon", tx2gene = tx2gene)
```

Convert read counts into DESeqDataSet

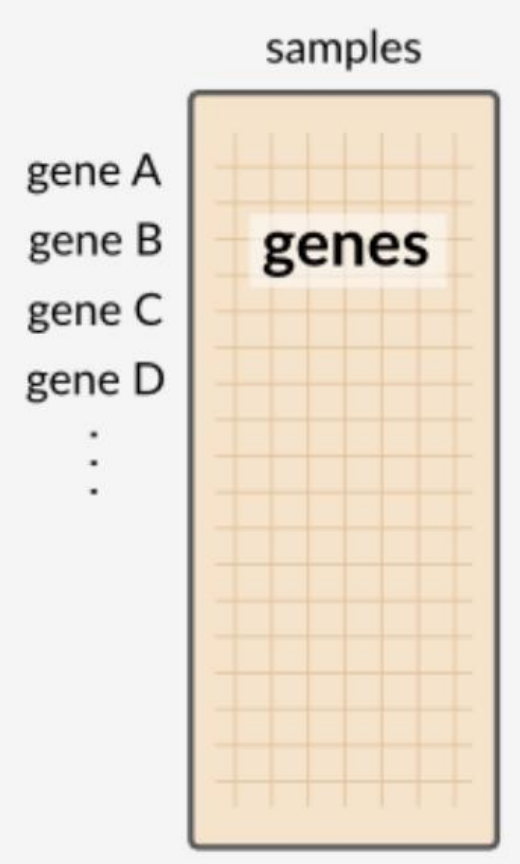



```
all(colnames(cts)==rownames(coldata))
```

```
library("DESeq2")

dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData = coldata,
                              design = ~ condition)
```

```
dds <- DESeqDataSetFromTximport(txi = txi,
                                colData = coldata,
                                design = ~ condition)
```

Convert read counts into DESeqDataSet




```
all(colnames(cts)==rownames(coldata))
```

```
library("DESeq2")

dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData = coldata,
                              design = ~ condition)
```

```
dds <- DESeqDataSetFromTximport(txi = txi,
                                colData = coldata,
                                design = ~ condition)
```

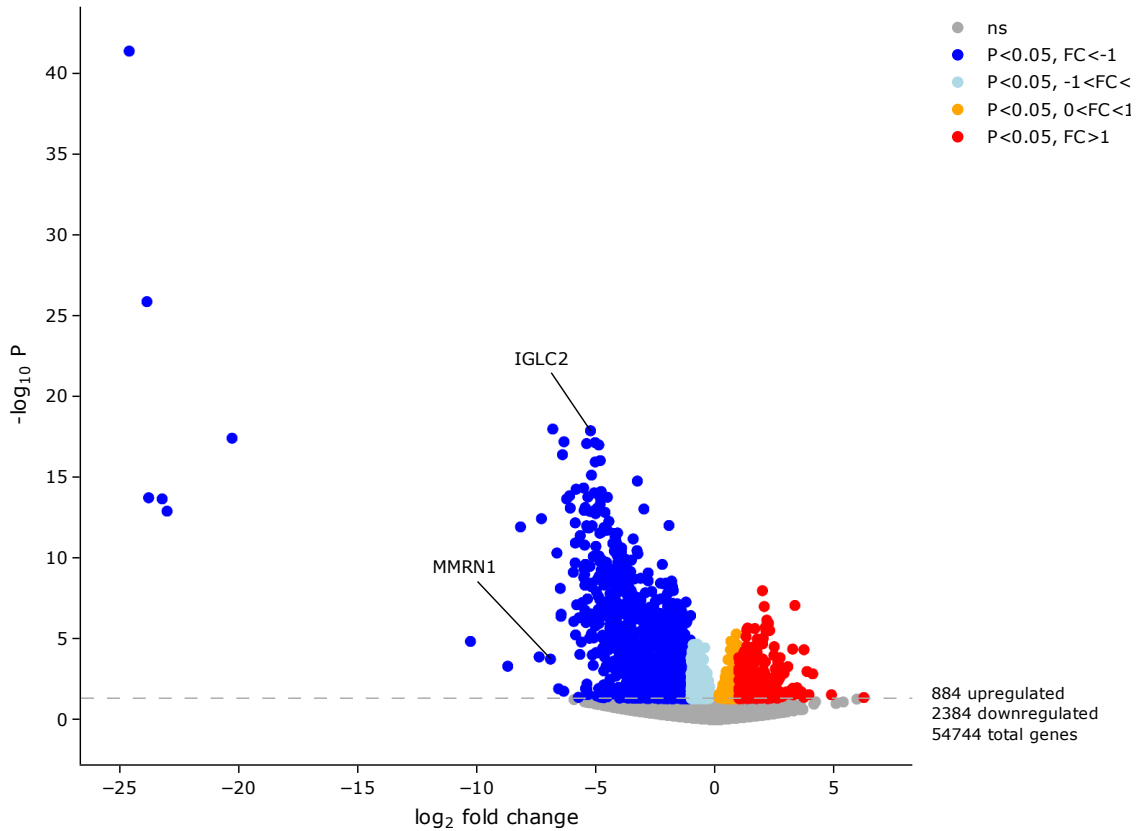
```
dds <- estimateSizeFactors(dds)
sizeFactors(dds)
dds <- estimateDispersions(dds)
dds <- nbinomWaldTest(dds)
dds <- DESeq(dds)
dssDF <- as.data.frame(results(dds, contrast=c("Genotype","KO","WT")))
```

Visualization of RNA-Seq results

- Volcano plot
- Heatmap
- Boxplot
- PCA for QC

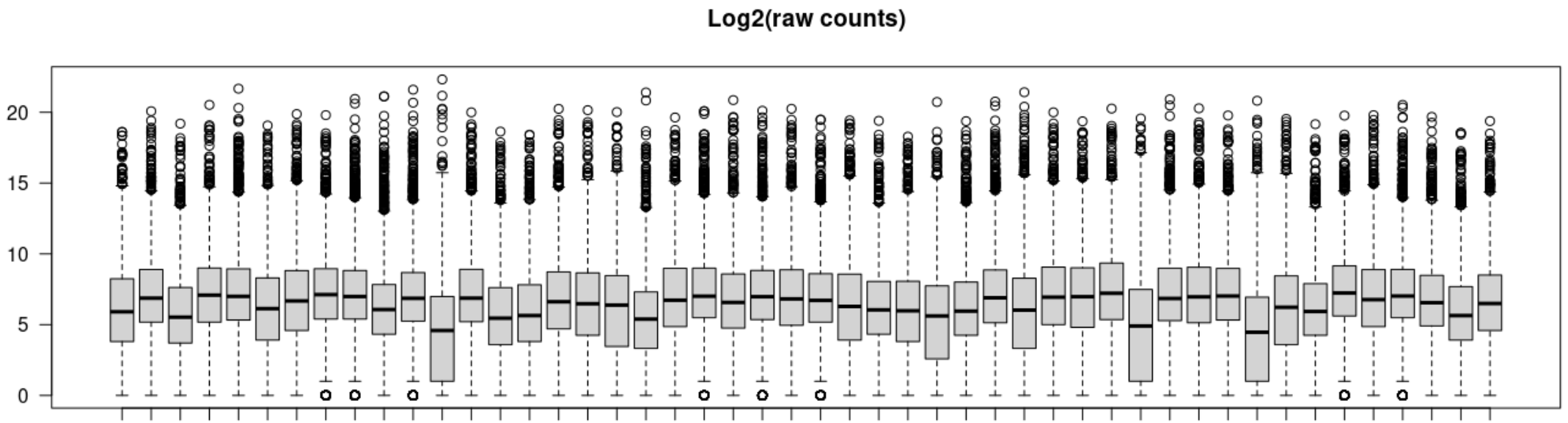
Volcano plot - easylabel

```
library("easylabel")
easyVolcano(data=dssDF, x = "log2FoldChange", y = "pvalue", padj = "padj")
easyVolcano(data=dssDF, x = "log2FoldChange", y = "pvalue")
```



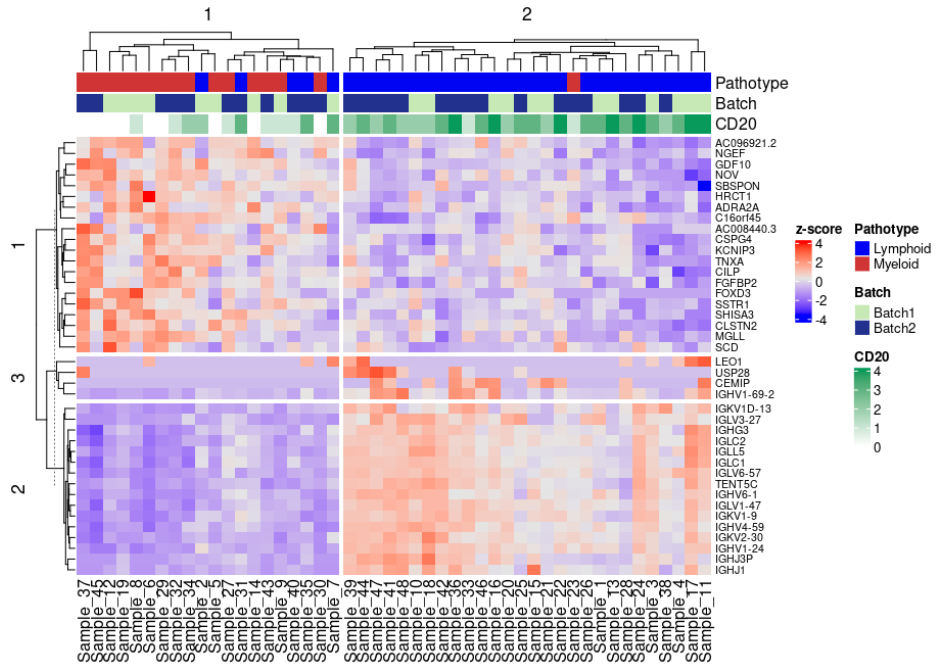
Count normalization with DESeq2

```
library("DESeq2")

vstCounts <- vst(dds, blind=T)
vstCounts <- assay(vst)

normCounts <- log2(counts(dds, normalized=TRUE) + 1)
```



Heatmap - ComplexHeatmap

```
library("ComplexHeatmap")

# Scale data before drawing heatmap

ScaledvstCounts <- t(scale(t(vstCounts)))


# Prepare annotation track
colBatch <- c("Batch1" = "#c7e9b6", "Batch2" = "#22318d")
colPatho <- c("Fibroid" = "#4bd950", "Myeloid" = "#d03435", "Lymphoid" = "#0004f3", "Ungraded" = "#bfc0bd")
column_ha = HeatmapAnnotation("Pathotype" = syn_metadata$Pathotype,
                    "Batch" = syn_metadata$Batch,
                    "CD20" = as.numeric(syn_metadata$CD20.max),

                     col = list("Pathotype" = colPatho, "Batch" = colBatch)
                    )

# Draw a heatmap
set.seed(123)
HM <- Heatmap(matrix = ScaledvstCounts,
              name = "z-score",
              cluster_rows = T,
              cluster_columns = T,
              column_km = 2,
              row_km = 3,
              col = c("blue","gray90","red"),
              row_names_gp = gpar(fontsize = 8),
              top_annotation = column_ha
             )
draw(HM)
```

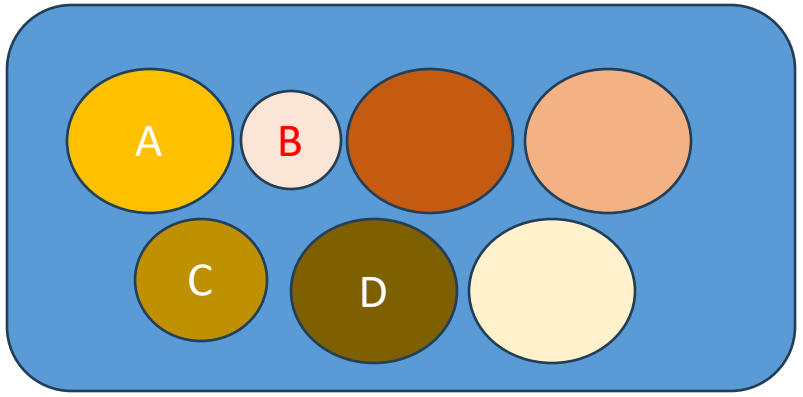


Geneset enrichment analysis

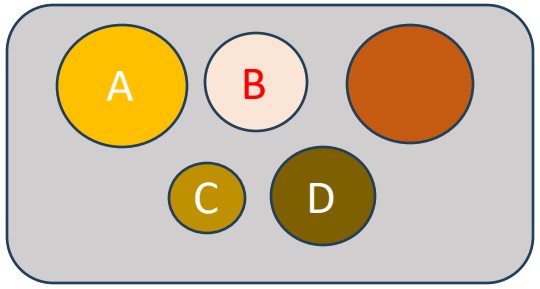
```
library("enrichR")
selectedDatabases <- c("KEGG_2021_Human","Reactome_2022","WikiPathway_2023_Human")
enriched <- enrichr(genes = rownames(dssDF)[which(dssDF$padj < 0.05),], databases = selectedDatabases)
                                                                    $pvalue
```

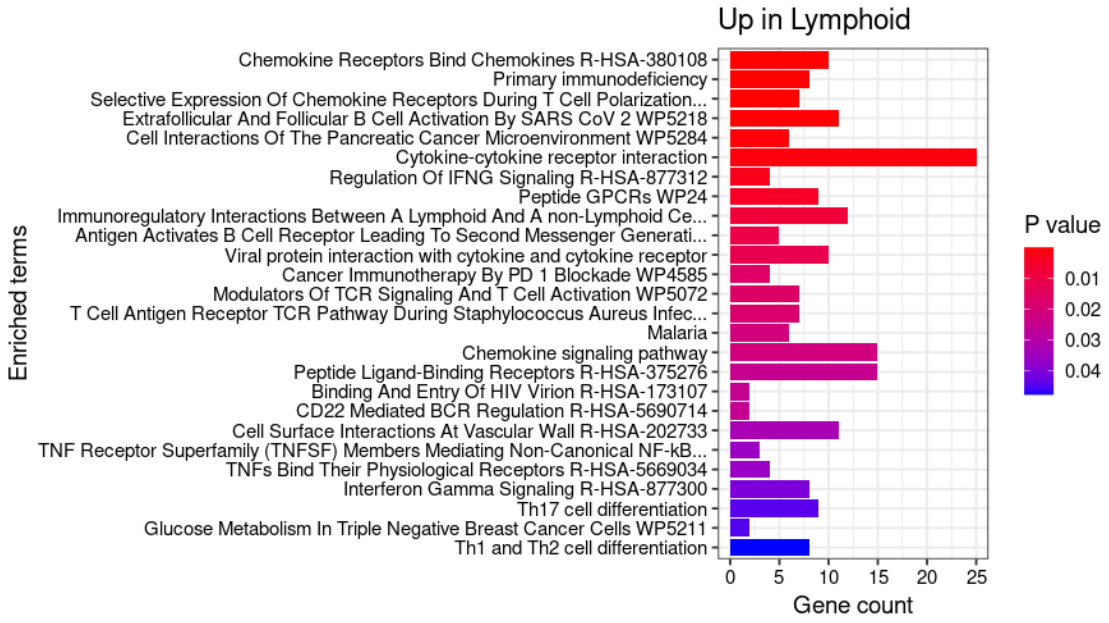
DEG list, FDR < 0.05

Genome



DEG list, p < 0.05





PCA for QC

```
# Filter genes by expression level
dim(vstCounts)
vst.pca <- vstCounts[which(!apply(counts(dds), 1, function(x) all(x<=50))),]
dim(vst.pca)

# Compute Run Principal Component Analysis
pc <- prcomp(t(vst.pca), scale. = T)


# Create PCA plot
library("ggplot2")
pc_df <- as.data.frame(pc$x[,c("PC1","PC2")])


pc_df$Batch <- syn_metadata$Batch
p <- ggplot(pc_df,aes(x=PC1,y=PC2, color=Batch))
p <- p + geom_point(size=3, alpha = 0.5)
p


# Repeat the analyis using Batch as a covariate in the design formula.
dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData = coldata,
                              design= ~ batch + condition)
```