rna_seq_workflow

Shaleigh Smith

9/14/2021

RNA-Seq Workflow

Transcriptomic Data processing

1. Data download from SRA.
2. Trimming and quality control with TrimGalore.
3. Alignment with Kallisto (example alignment for STAR and BBmap included).
4. Quantification of kallisto with tximport.
5. Differential expression analysis with DESeq2.

Data

Whole genome sequencing & RNA sequencing of human well differentiated liposarcoma

Select all samples for SRA selector:

- SRR15320001
- SRR15320004
- SRR15320005
- SRR15320006
- SRR15320007
- SRR15320009
- SRR15320010
- SRR15320008
- SRR15320012

- SRR15320011
- SRR15320002

```
# Download human genome
# https://support.illumina.com/sequencing/sequencing_software/igenome.html
cd ~/Desktop/
pwd
wget http://igenomes.illumina.com.s3-website-us-east-1.amazonaws.com/Homo_sapiens/UCSC/hg38/Homo_sapiens
tar -zxvf Homo_sapiens_UCSC_hg38.tar.gz
```

```
# GCC path if needed for install: /usr/local/Cellar/gcc/11.2.0/bin/g++-11

# If using STAR for sequence alignment...
# Can only do this and map with 30GB+ of RAM - must use cluster

## Create STAR genome index
STAR --runThreadN 1 \
--runMode genomeGenerate \
--genomeDir ~/Desktop/work/rna_seq/data/star_hg38_index \
--genomeFastaFiles ~/Desktop/work/rna_seq/data/Homo_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.
--sjdbGTFfile ~/Desktop/work/rna_seq/data/Homo_sapiens/UCSC/hg38/Annotation/Genes/genes.gtf \
--sjdbOverhang 99

# Iterate through project Run IDs
cd ./data/fastq/
pwd
while IFS= read -r i
  do
  echo $i

  # Get SRA IDs for Liposarcoma samples
  # Download data from SRA, put each read into separate files (paired data)
  /Users/sha/sratoolkit.2.11.1-mac64/bin/fastq-dump --accession $i --split-files --gzip -O ./

  # Can also download manually here:
  # https://www.ebi.ac.uk/ena/browser/view/SRR15320006?show=reads

  # Trim and run FASTQC
  echo ${i}_1.fastq.gz
  echo ${i}_2.fastq.gz
  ~/TrimGalore-0.6.6/trim_galore --path_to_cutadapt  ~/.local/bin/cutadapt --q 30  --phred33  -o ./  --

  # Align reads to reference genome with STAR (on cluster)
  # Spliced Transcripts Alignment to a Reference (STAR) is a fast RNA-seq read mapper...
  # with support for splice-junction and fusion read detection.
  # https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf
  # MAPQ=255 is defaut (uniquely mapping reads)
  # Convert sam to bam and sort
  STAR --genomeDir ~/Desktop/work/rna_seq/data/star_hg38_index/ \
  --runThreadN 1 \
  --readFilesIn ${i}_1_val_1.fq.gz ${i}_2_val_2.fq.gz \
  --outSAMtype BAM SortedByCoordinate \
  --outSAMunmapped Within

  # Create a feature counts matrix
  # The input is specified as reversely stranded (-s 2) and paired end (-p).
  # The fragment length is checked (-P)
  # Only the fragments that have both ends successfully aligned are counted (-B)
  # Chimeric fragments are ignored (-C) as are duplicates (--ignoreDup)
  # Only primary alignments are counted (--primary)
  # Counts are based on gene_id (-g).
  /Users/sha/subread-2.0.3-source/bin/featureCounts -s 2 -p -B -C -P  --ignoreDup  --primary  -a  ../Hon

done < ../SRR_Acc_List.txt
```

```
# Build BBMap index
/Users/sha/bbmap/bbmap.sh -Xmx20G ref=../Homo_sapiens_UCSC_hg38/Homo_sapiens/UCSC/hg38/Sequence/WholeGen

# Iterate through project Run IDs
cd ./data/fastq/
pwd
while IFS= read -r i
  do
  echo $i

  # Get SRA IDs for Liposarcoma samples
  # Download data from SRA, put each read into separate files (paired data)
  /Users/sha/sratoolkit.2.11.1-mac64/bin/fastq-dump --accession $i --split-files --gzip -O ./

  # Can also download manually here:
  # https://www.ebi.ac.uk/ena/browser/view/SRR15320006?show=reads

  # Trim and run FASTQC
  echo ${i}_1.fastq.gz
  echo ${i}_2.fastq.gz
  ~/TrimGalore-0.6.6/trim_galore --path_to_cutadapt  ~/.local/bin/cutadapt --q 30  --phred33  -o ./  --p

  # Align read to reference genome with BBMap
  # BBmap is splice-aware but uses less RAM than STAR
  # minid=0.76: Approximate minimum alignment identity to look for. Higher is faster and less sensitive
  # ambiguous: Set behavior on ambiguously-mapped reads (with multiple top-scoring mapping locations).
  bbmap.sh -Xmx6G path=../bbmap_index/ in=SRR15320001_1_val_1.fq.gz  in2=SRR15320001_2_val_2.fq.gz  outm

  # Convert SAM file to BAM file and sort
  samtools view -S -b ../align/${i}.sam > ../align/${i}.bam
  samtools sort ../align/${i}.bam -o ../align/${i}_sorted.bam

  # Create a feature counts matrix
  # The input is specified as reversely stranded (-s 2) and paired end (-p).
  # The fragment length is checked (-P)
  # Only the fragments that have both ends successfully aligned are counted (-B)
  # Chimeric fragments are ignored (-C) as are duplicates (--ignoreDup)
  # Only primary alignments are counted (--primary)
  # Counts are based on gene_id (-g).
  /Users/sha/subread-2.0.3-source/bin/featureCounts -s 2 -p -B -C -P  --ignoreDup  --primary  -a  ../Hom

done < ../SRR_Acc_List.txt
```

```
# Create kallisto index
# Must build from homo sapiens coding sequences (transcriptome of organism of interest)
# Can download from Ensembl FTP Site: http://useast.ensembl.org/info/data/ftp/index.html
kallisto index --index=Homo_sapiens_GRCh38_cds_all_kallisto_index Homo_sapiens.GRCh38.cds.all.fa

# Iterate through project Run IDs
cd ./data/fastq/
pwd
while IFS= read -r i
  do
  echo $i

  # Get SRA IDs for Liposarcoma samples
  # Download data from SRA, put each read into separate files (paired data)
  /Users/sha/sratoolkit.2.11.1-mac64/bin/fastq-dump --accession $i --split-files --gzip -O ./

  # Can also download manually here:
  # https://www.ebi.ac.uk/ena/browser/view/SRR15320006?show=reads

  # Trim and run FASTQC
  echo ${i}_1.fastq.gz
  echo ${i}_2.fastq.gz
  ~/TrimGalore-0.6.6/trim_galore --path_to_cutadapt  ~/.local/bin/cutadapt --q 30  --phred33  -o ./  --p

  # Run kallisto
  # Get abundance estimates through pseudoalignment
  kallisto quant -i ../kallisto/Homo_sapiens_GRCh38_cds_all_kallisto_index -o ../align/${i} -b 100 --bia

done < ../SRR_Acc_List.txt
```

Differential Expression Analysis

```
# Library
library(tidyverse)
library(biomaRt)
library(tximport)
library(rhdf5)
library(DESeq2)
```

```
## Warning: package 'BiocGenerics' was built under R version 4.0.5

## Warning: package 'GenomeInfoDb' was built under R version 4.0.5
```

```
library(ComplexHeatmap)
library(circlize)
library(AnnotationDbi)
library(org.Hs.eg.db)
library(ggrepel)
```

```
# Import ensembla annotations for the human genome
mart <- biomaRt::useMart(biomart = "ensembl", dataset =  "hsapiens_gene_ensembl")
bio_mart_df <- biomaRt::getBM(mart = mart,
                         attributes = c("ensembl_transcript_id", "transcript_version", "ensembl_gene_id
                                        "external_gene_name", "description", "transcript_biotype",
                                        "gene_biotype"))

# Combine transcript id and version
bio_df <- bio_mart_df %>%
  mutate(target_id = paste0(ensembl_transcript_id, ".", transcript_version)) %>%
  dplyr::rename(gene_symbol = external_gene_name,
                full_name = description,
                biotype = transcript_biotype) %>%
  dplyr::select(-ensembl_transcript_id, -transcript_version)
head(bio_df)
```

```
##   ensembl_gene_id gene_symbol
## 1 ENSG00000210049       MT-TF
## 2 ENSG00000211459     MT-RNR1
## 3 ENSG00000210077       MT-TV
## 4 ENSG00000210082     MT-RNR2
## 5 ENSG00000209082      MT-TL1
## 6 ENSG00000198888      MT-ND1
##                                                                                              full
## 1                              mitochondrially encoded tRNA-Phe (UUU/C) [Source:HGNC Symbol;Acc:HGNC
## 2                                      mitochondrially encoded 12S rRNA [Source:HGNC Symbol;Acc:HGNC
## 3                                mitochondrially encoded tRNA-Val (GUN) [Source:HGNC Symbol;Acc:HGNC
## 4                                      mitochondrially encoded 16S rRNA [Source:HGNC Symbol;Acc:HGNC
## 5                            mitochondrially encoded tRNA-Leu (UUA/G) 1 [Source:HGNC Symbol;Acc:HGNC
## 6 mitochondrially encoded NADH:ubiquinone oxidoreductase core subunit 1 [Source:HGNC Symbol;Acc:HGNC
##          biotype   gene_biotype         target_id
## 1        Mt_tRNA        Mt_tRNA ENST00000387314.1
## 2        Mt_rRNA        Mt_rRNA ENST00000389680.2
## 3        Mt_tRNA        Mt_tRNA ENST00000387342.1
## 4        Mt_rRNA        Mt_rRNA ENST00000387347.2
## 5        Mt_tRNA        Mt_tRNA ENST00000386347.1
## 6 protein_coding protein_coding ENST00000361390.2
```

```
# Convert kallisto abundance.tsv (or abundance.h5) files into a gene count format for analysis using ei
# Use tximport to summarize transcript counts into gene counts
kallisto_names <- list.dirs("/Users/sha/Desktop/work/rna_seq/data/align/",  full.names = FALSE)[-1]
kallisto_dir <- list.dirs("/Users/sha/Desktop/work/rna_seq/data/align/", full.names = TRUE)[-1]
kallisto_files <- file.path(kallisto_dir,"abundance.tsv")
names(kallisto_files) <- kallisto_names
tx_df <- tximport(kallisto_files,
                  type = "kallisto",
                  tx2gene = dplyr::select(bio_df, target_id, gene_symbol),
                  countsFromAbundance ="no")
```

```
## Note: importing `abundance.h5` is typically faster than `abundance.tsv`

## reading in files with read_tsv

## 1 2 3 4 5 6 7 8 9 10 11
## summarizing abundance
## summarizing counts
## summarizing length
```

```
summary(tx_df)
```

```
##                     Length Class  Mode
## abundance           218141 -none- numeric
## counts              218141 -none- numeric
## length              218141 -none- numeric
## countsFromAbundance      1 -none- character
```

```
# Subset output for protein coding genes only
# Round all values (DESEQ does not like fractions/decimals)
count_df <- tx_df$counts %>%
  as.data.frame() %>%
  tibble::rownames_to_column("gene_symbol") %>%
  filter(gene_symbol %in% filter(bio_df, gene_biotype == "protein_coding")$gene_symbol &
           gene_symbol != "") %>%
  mutate_if(is.numeric, round) %>%
  tibble::column_to_rownames("gene_symbol")
head(count_df)
```

```
##           SRR15320001 SRR15320002 SRR15320004 SRR15320005 SRR15320006 SRR15320007
## A1BG                1           0           3           0           5           2
## A1CF                0           0           2           0           0           0
## A2M               117         225         995        2210        2283         462
## A2ML1               1           0           3           0           2           5
## A3GALT2             0           0           2           0           2           0
## A4GALT              7         178          16          10          12           3
##           SRR15320008 SRR15320009 SRR15320010 SRR15320011 SRR15320012
## A1BG                2           2           0           1           0
## A1CF                0           0           5           0           0
## A2M              1253         648        1469         121         495
## A2ML1               3           4           1           1           0
## A3GALT2             1           0           1           2           6
## A4GALT             13          22          19          18          26
```

```
# Save
#write.table(count_df, "./data/count/kallisto_counts.txt", sep = "\t", row.names = T, quote = F)
```

```r
# Create sample annotation df for conditions input
clin_df <- read.delim("./data/SraRunTable.txt", sep = ",")
clin_df <- clin_df %>%
  dplyr::select(Run, tissue) %>%
  arrange(Run) %>%
  mutate_if(is.character, ~ gsub(" ", "_", .)) %>%
  mutate(tissue = factor(tissue, levels = c("Normal_fat","Liposarcoma")))
clin_df
```

```
##            Run      tissue
## 1  SRR15320001 Liposarcoma
## 2  SRR15320002  Normal_fat
## 3  SRR15320004 Liposarcoma
## 4  SRR15320005 Liposarcoma
## 5  SRR15320006  Normal_fat
## 6  SRR15320007 Liposarcoma
## 7  SRR15320008  Normal_fat
## 8  SRR15320009 Liposarcoma
## 9  SRR15320010 Liposarcoma
## 10 SRR15320011 Liposarcoma
## 11 SRR15320012 Liposarcoma
```

```r
# Read in count data
count_df <- read.delim("./data/count/kallisto_counts.txt")

# Check order
clin_df$Run == colnames(count_df) #TRUE
```

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

```r
# Create DESeq2 data set
de_data_df <- DESeqDataSetFromMatrix(countData = count_df,
                                     colData = clin_df,
                                     design = ~ tissue)
head(de_data_df)
```

```
## class: DESeqDataSet
## dim: 6 11
## metadata(1): version
## assays(1): counts
## rownames(6): A1BG A1CF ... A3GALT2 A4GALT
## rowData names(0):
## colnames(11): SRR15320001 SRR15320002 ... SRR15320011 SRR15320012
## colData names(2): Run tissue
```

```r
# Remove genes that do not have counts greater than 2 in at least 2 of the samples (columns)
dim(de_data_df)
```

```
## [1] 19379    11
```

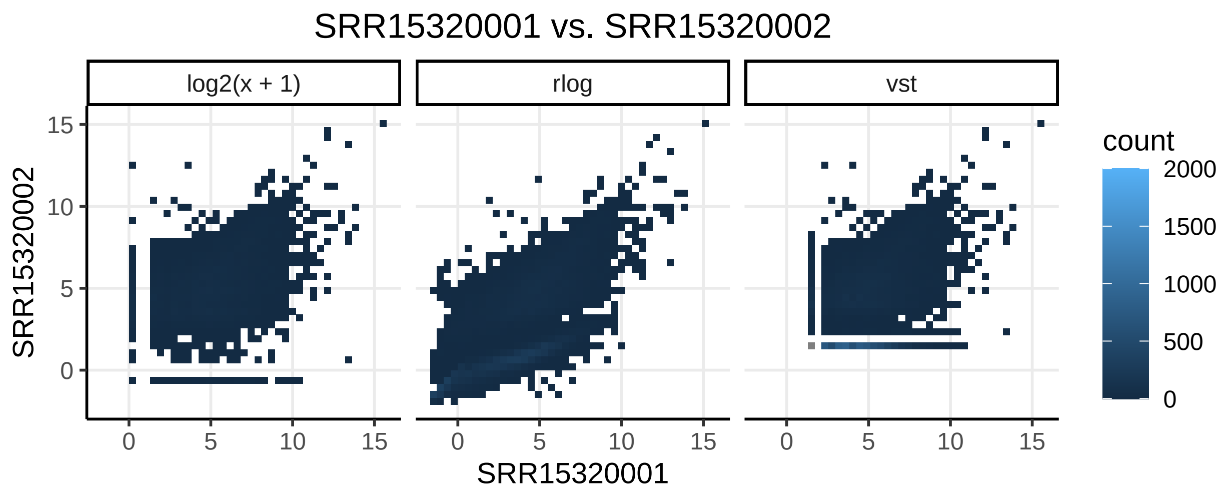
```r
de_data_df <- de_data_df[rowSums(counts(de_data_df) >= 2) >= 2]
dim(de_data_df)
```

```
## [1] 15805    11
```

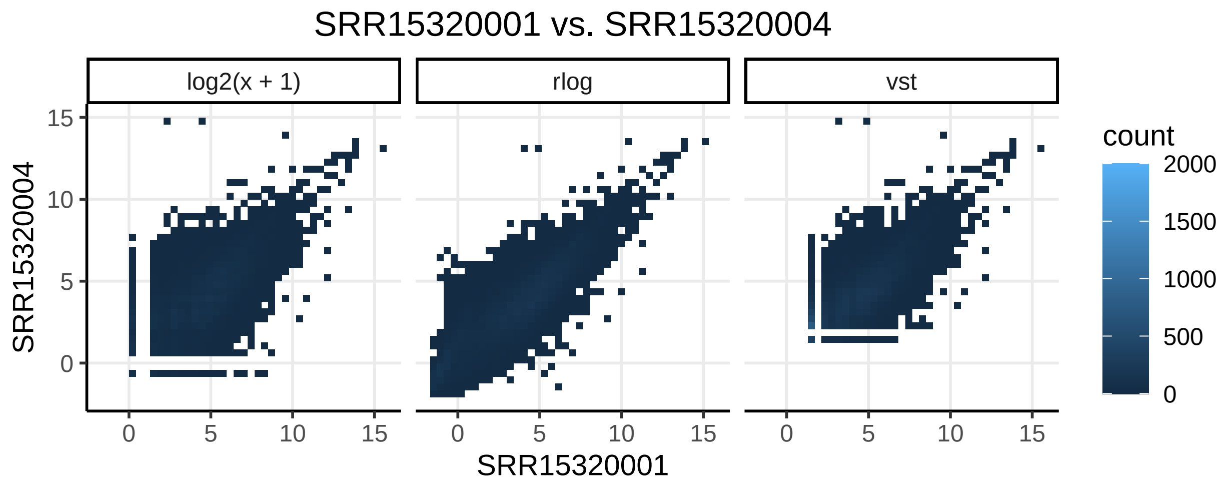
```r
# Confirm that all samples are labelled correctly
as.data.frame(colData(de_data_df))
```

```
##                     Run      tissue
## SRR15320001 SRR15320001 Liposarcoma
## SRR15320002 SRR15320002  Normal_fat
## SRR15320004 SRR15320004 Liposarcoma
## SRR15320005 SRR15320005 Liposarcoma
## SRR15320006 SRR15320006  Normal_fat
## SRR15320007 SRR15320007 Liposarcoma
## SRR15320008 SRR15320008  Normal_fat
## SRR15320009 SRR15320009 Liposarcoma
## SRR15320010 SRR15320010 Liposarcoma
## SRR15320011 SRR15320011 Liposarcoma
## SRR15320012 SRR15320012 Liposarcoma
```

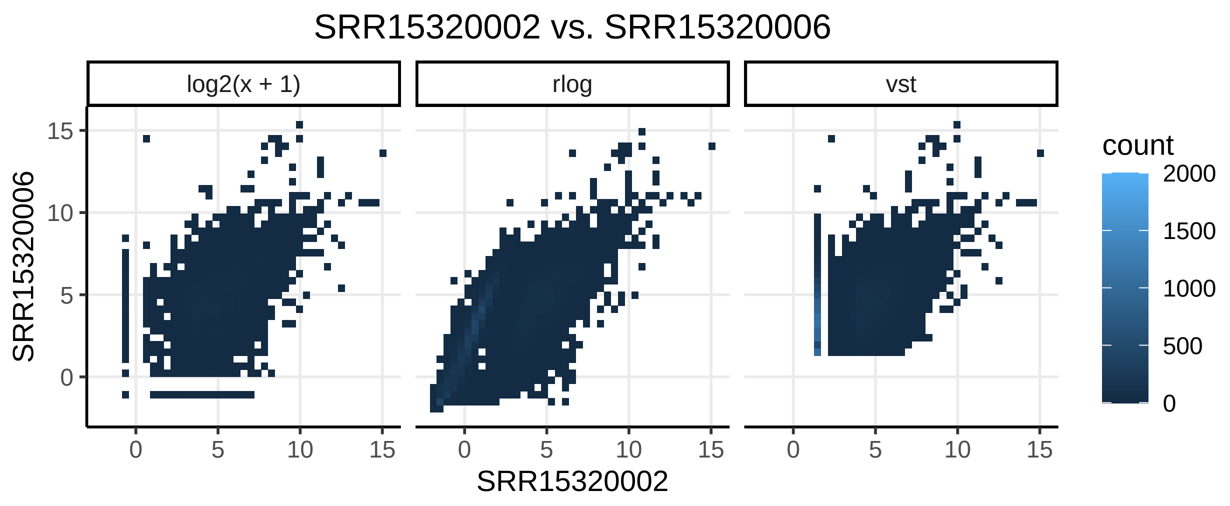
```r
### Exploratory data analysis of DESeq2 matrix with transformation

# Estimate size factors
# The size factor is the median ratio of the sample over a pseudosample: for each gene, the geometric m
deseq_eda <- estimateSizeFactors(de_data_df)

# Apply regularized-logarithm transformation
rld <- rlog(deseq_eda, blind = FALSE)

# Apply variance stabilizing transformation
vsd <- vst(deseq_eda, blind = FALSE)

# Create new data frame three normalization methods for all samples
deseq_eda <- bind_rows(
  as_data_frame(log2(counts(deseq_eda, normalized=TRUE)[, (1:8)])) %>%
         mutate(transform = "log2(x + 1)"),
  as_data_frame(assay(vsd)[, (1:8)]) %>% mutate(transform = "vst"),
  as_data_frame(assay(rld)[, (1:8)]) %>% mutate(transform = "rlog"))

# Compare transformation visually
ggplot(deseq_eda, aes(x = SRR15320001, y = SRR15320002)) +
  geom_bin2d(bins = 40) +
  coord_fixed() +
  facet_grid( . ~ transform) +
  scale_fill_continuous(limits = c(0, 2000), breaks = seq(0, 2000, by = 500)) +
  ggtitle("SRR15320001 vs. SRR15320002")+
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line())
```



```r
ggplot(deseq_eda, aes(x = SRR15320001, y = SRR15320004)) +
  geom_bin2d(bins = 40) +
  coord_fixed() +
  facet_grid( . ~ transform) +
  scale_fill_continuous(limits = c(0, 2000), breaks = seq(0, 2000, by = 500)) +
  ggtitle("SRR15320001 vs. SRR15320004") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line())
```



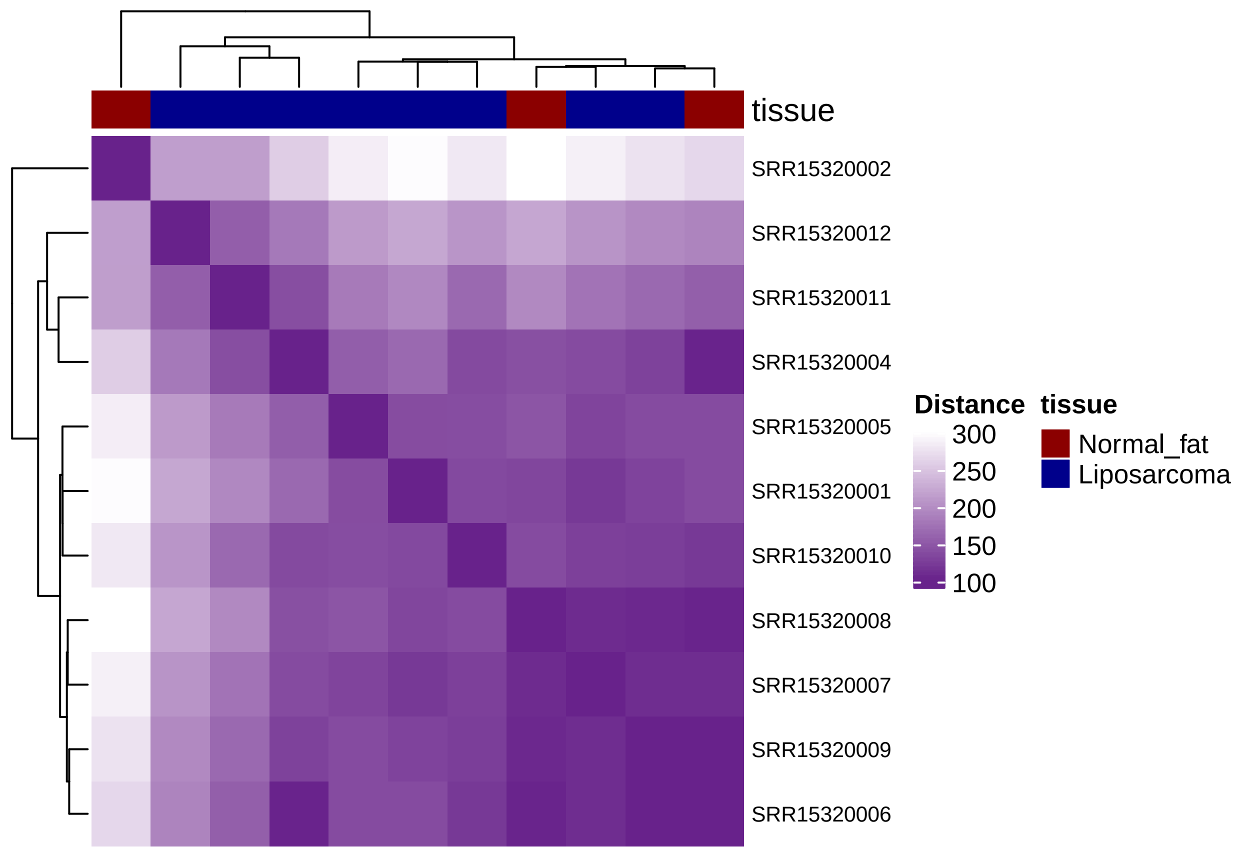
```
ggplot(deseq_eda, aes(x = SRR15320002, y = SRR15320006)) +
  geom_bin2d(bins = 40) +
  coord_fixed() +
  facet_grid( . ~ transform) +
  scale_fill_continuous(limits = c(0, 2000), breaks = seq(0, 2000, by = 500)) +
  ggtitle("SRR15320002 vs. SRR15320006") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line())
```



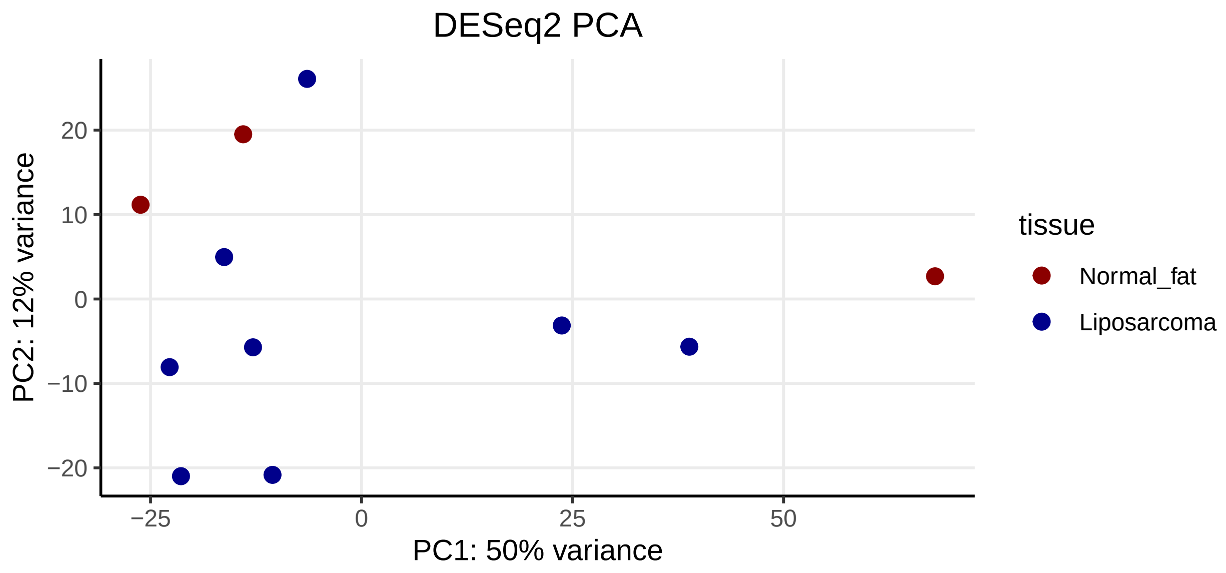
```
### Exploratory data analysis of DESeq matrix with sample comparison
# This will asses overall similarity between samples
# As shown in the above plots this similarity might not be as expected

# Calculate the euclidean distance between samples
deseq_distance <- dist(t(assay(rld)))
deseq_distance
```

```
##             SRR15320001 SRR15320002 SRR15320004 SRR15320005 SRR15320006
## SRR15320002    297.7160
## SRR15320004    164.0792    256.4884
## SRR15320005    137.2563    284.9863    153.4614
## SRR15320006    136.2246    265.3061    101.0151    136.1942
## SRR15320007    119.2113    286.8889    136.0561    129.8038    109.1608
## SRR15320008    131.1255    300.2149    140.6991    145.2741    101.2123
## SRR15320009    128.9992    274.5189    127.2853    136.0384    100.0010
## SRR15320010    134.0189    280.1184    134.9080    137.5094    119.1829
## SRR15320011    193.0612    213.5031    138.8931    180.6102    154.5112
## SRR15320012    221.5021    213.8302    179.0983    210.0676    189.2476
##             SRR15320007 SRR15320008 SRR15320009 SRR15320010 SRR15320011
## SRR15320002
## SRR15320004
## SRR15320005
## SRR15320006
## SRR15320007
## SRR15320008    107.3435
## SRR15320009    108.5663    104.5714
## SRR15320010    126.0245    135.9556    123.8325
## SRR15320011    173.9061    194.3734    164.4699    164.5721
## SRR15320012    204.4379    221.0412    194.2668    205.2472    154.2133
```

```
# Create annotation for the heatmap
deseq_ha <- HeatmapAnnotation(df = dplyr::select(clin_df, tissue),
                              col = list(tissue = c("Normal_fat" = "darkred",
                                                    "Liposarcoma" = "darkblue")))

# Visualize with heatmap
deseq_heat <- Heatmap(as.matrix(deseq_distance),
                      top_annotation = deseq_ha,
                      show_column_names = F,
                      show_row_names = T,
                      cluster_rows = T,
                      cluster_columns = T,
                      clustering_method_columns = 'complete',
                      clustering_method_rows = "complete",
                      row_names_gp = gpar(fontsize = 8),
                      heatmap_legend_param = list(title = "Distance"),
                      col = colorRamp2(c(min(deseq_distance), max(deseq_distance)), c("darkorchid4", "white

deseq_heat
```



```r
### Compare with PCA
# Create PCA object to be plotted with ggplot
plot_pca <- plotPCA(rld, intgroup = c("tissue"), returnData = TRUE)

# Calculate and round variance for plotting
percentVar <- round(100 * attr(plot_pca, "percentVar"))

# Plot PCA
deseq_pca_1 <- ggplot(plot_pca, aes(x = PC1, y = PC2, color = tissue)) +
  geom_point(size = 2.5) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  scale_color_manual(values = c("darkred", "darkblue")) +
  ggtitle("DESeq2 PCA") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line())

deseq_pca_1
```



```r
# Make sure that normal is the first level in the sample factor
# This is to ensure the log2 fold change is calculated over the control (when results are called at ran
levels(de_data_df$tissue)
```

```
## [1] "Normal_fat"  "Liposarcoma"
```

```
# Check to insure all samples are correct
as.data.frame(colData(de_data_df))
```

```
##                     Run      tissue
## SRR15320001 SRR15320001 Liposarcoma
## SRR15320002 SRR15320002  Normal_fat
## SRR15320004 SRR15320004 Liposarcoma
## SRR15320005 SRR15320005 Liposarcoma
## SRR15320006 SRR15320006  Normal_fat
## SRR15320007 SRR15320007 Liposarcoma
## SRR15320008 SRR15320008  Normal_fat
## SRR15320009 SRR15320009 Liposarcoma
## SRR15320010 SRR15320010 Liposarcoma
## SRR15320011 SRR15320011 Liposarcoma
## SRR15320012 SRR15320012 Liposarcoma
```

```
# Run the DESeq analysis against raw counts
de_final_df <- DESeq(de_data_df)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 485 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```
results(de_final_df)
```

```
## log2 fold change (MLE): tissue Liposarcoma vs Normal fat
## Wald test p-value: tissue Liposarcoma vs Normal fat
## DataFrame with 15805 rows and 6 columns
##           baseMean log2FoldChange     lfcSE      stat    pvalue      padj
##          <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
## A1BG      1.302020      -0.314635  1.535480 -0.204910  0.837642        NA
## A1CF      0.503052       2.545968  3.499705  0.727481  0.466931        NA
## A2M     770.970491      -0.136318  0.753981 -0.180797  0.856527  0.999876
## A2ML1     1.783860       0.729162  1.419215  0.513778  0.607407        NA
## A3GALT2   1.510469       1.402293  1.892471  0.740986  0.458702        NA
## ...            ...            ...       ...       ...       ...       ...
## ZXDC       26.3745     -0.4457435  0.572139 -0.779083  0.435931  0.999876
## ZYG11B     36.7254     -0.1022934  0.999063 -0.102389  0.918448  0.999876
## ZYX       182.8157      0.5199805  0.693291  0.750018  0.453244  0.999876
## ZZEF1     165.0703     -0.0327483  0.363401 -0.090116  0.928195  0.999876
## ZZZ3       51.2117     -0.1402562  0.787024 -0.178211  0.858557  0.999876
```

```
# Call results
# Contrast: a character vector with exactly three elements:
# 1. the name of a factor in the design formula
# 2. the name of the numerator level for the fold change
# 3. the name of the denominator level for the fold change (simplest case)
de_res_df <- results(de_final_df, contrast = c("tissue", "Liposarcoma", "Normal_fat"))
de_res_df
```

```
## log2 fold change (MLE): tissue Liposarcoma vs Normal_fat
## Wald test p-value: tissue Liposarcoma vs Normal fat
## DataFrame with 15805 rows and 6 columns
##           baseMean log2FoldChange     lfcSE      stat    pvalue      padj
##          <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
## A1BG      1.302020      -0.314635  1.535480 -0.204910  0.837642        NA
## A1CF      0.503052       2.545968  3.499705  0.727481  0.466931        NA
## A2M     770.970491      -0.136318  0.753981 -0.180797  0.856527  0.999876
## A2ML1     1.783860       0.729162  1.419215  0.513778  0.607407        NA
## A3GALT2   1.510469       1.402293  1.892471  0.740986  0.458702        NA
## ...            ...            ...       ...       ...       ...       ...
## ZXDC       26.3745     -0.4457435  0.572139 -0.779083  0.435931  0.999876
## ZYG11B     36.7254     -0.1022934  0.999063 -0.102389  0.918448  0.999876
## ZYX       182.8157      0.5199805  0.693291  0.750018  0.453244  0.999876
## ZZEF1     165.0703     -0.0327483  0.363401 -0.090116  0.928195  0.999876
## ZZZ3       51.2117     -0.1402562  0.787024 -0.178211  0.858557  0.999876
```

```
# Review Comparisons
mcols(de_res_df, use.names = TRUE)
```

```
## DataFrame with 6 rows and 2 columns
##                        type            description
##                 <character>            <character>
## baseMean       intermediate mean of normalized c..
## log2FoldChange      results log2 fold change (ML..
## lfcSE               results standard error: tiss..
## stat                results Wald statistic: tiss..
## pvalue              results Wald test p-value: t..
## padj                results   BH adjusted p-values
```

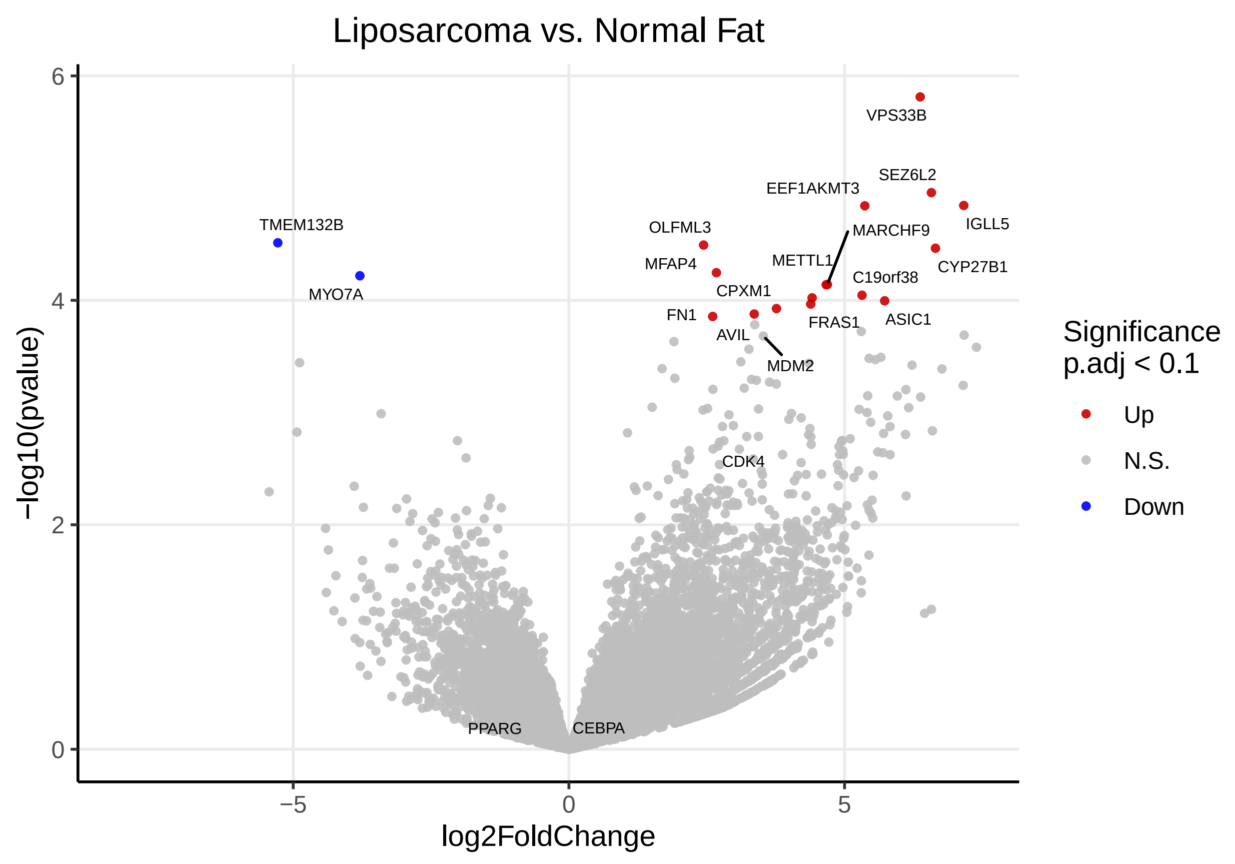
```
### Review Summary
summary(de_res_df, alpha = 0.01)
```

```
##
```

```
## out of 15805 with nonzero total read count
## adjusted p-value < 0.01
## LFC > 0 (up)       : 0, 0%
## LFC < 0 (down)     : 0, 0%
## outliers [1]       : 258, 1.6%
## low counts [2]     : 2739, 17%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
# Save
write.table(as.data.frame(de_res_df) %>% tibble::rownames_to_column("gene_symbol"),
            "./deseq2_liposarcoma_results.txt", sep = "\t", row.names = F, quote = F)
```

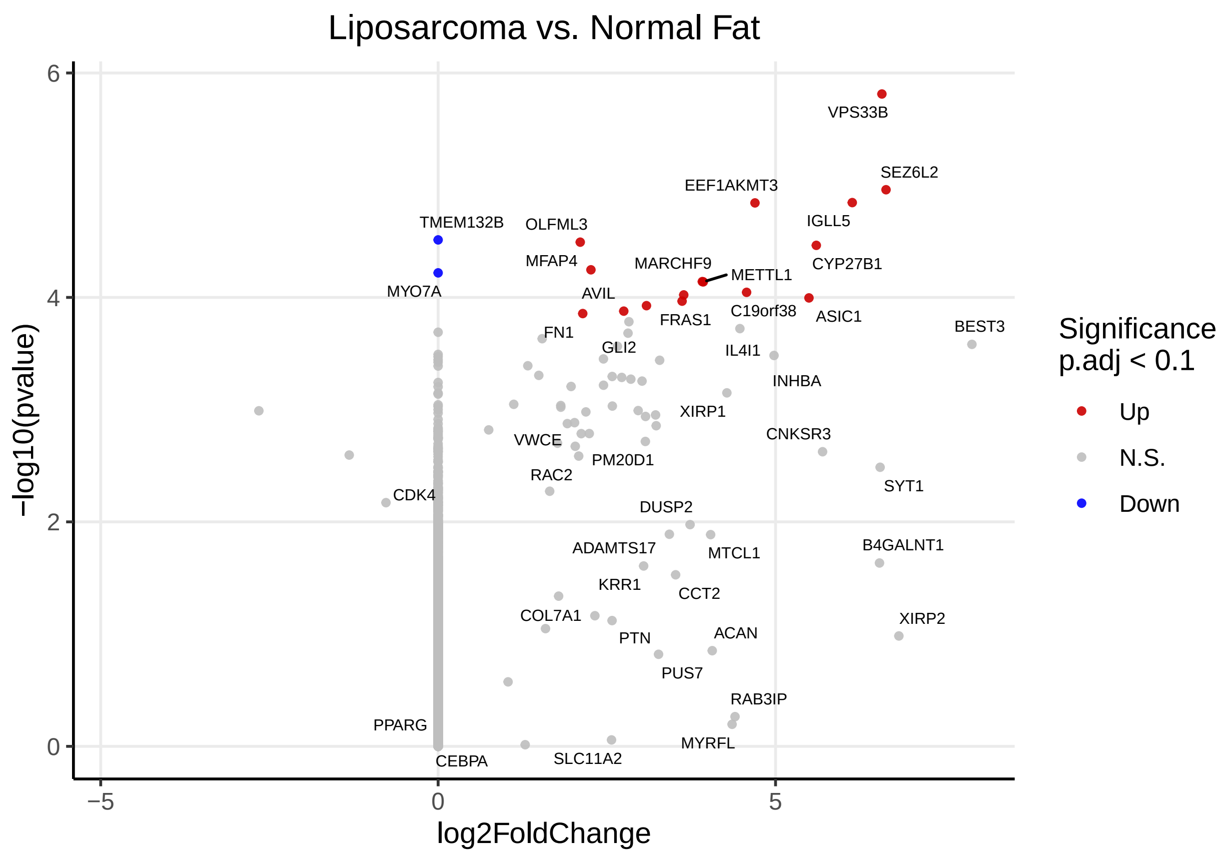
```
# Visualize significance
# Use adjusted p value threshold of 0.1 due to low number of significanlty regulated genes
# Also indicate genes of interest
vol_df <- as.data.frame(de_res_df) %>%
  tibble::rownames_to_column("gene_symbol") %>%
  mutate(threshold = case_when(log2FoldChange > 0 & padj < 0.1 ~ "up_sig",
                               log2FoldChange < 0 & padj < 0.1 ~ "down_sig",
                               TRUE ~ "not_sig"))

# Volcano plot
vol_plot <- ggplot(vol_df, aes(x = log2FoldChange, y = -log10(pvalue))) +
  geom_point(aes(col = threshold), size = 1, alpha = 0.9) +
  ggtitle("Liposarcoma vs. Normal Fat") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line()) +
  scale_colour_manual(values = c("red3", "grey", "blue", "black"),
                      breaks = c("up_sig", "not_sig", "down_sig", "other"),
                      labels = c("Up", "N.S.", "Down", "Other")) +
  geom_text_repel(data = filter(vol_df, padj < 0.1 | gene_symbol %in% c("CDK4", "MDM2", "PPARG", "CEBPA
                  aes(label = gene_symbol), size = 2) +
  labs(col = "Significance\np.adj < 0.1")
vol_plot
```



```
# Apply shrinkage to logFC
# See difference in visualization

# Calculate row means across samples
mean_norm <-  rowMeans(counts(de_final_df, normalized=TRUE)[, de_final_df$tissue == "Liposarcoma"])
mean_tumor <-  rowMeans(counts(de_final_df, normalized=TRUE)[, de_final_df$tissue == "Normal_fat"])

### Normal vs. Tumor
# Log Fold change
lfc_comp <- lfcShrink(de_final_df, coef = "tissue_Liposarcoma_vs_Normal_fat", 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 row names to log fold change data frame
lfc_comp <-  cbind(as.data.frame(lfc_comp), mean_norm, mean_tumor)

# Visualize significance
# Use adjusted p value threshold of 0.1 due to low number of significanlty regulated genes
# Also indicate genes of interest
lfc_vol_df <- lfc_comp %>%
  tibble::rownames_to_column("gene_symbol") %>%
  mutate(threshold = case_when(log2FoldChange > 0 & padj < 0.1 ~ "up_sig",
                                log2FoldChange < 0 & padj < 0.1 ~ "down_sig",
                                TRUE ~ "not_sig"))

# Volcano plot
lfc_vol_plot <- ggplot(lfc_vol_df, aes(x = log2FoldChange, y = -log10(pvalue))) +
  geom_point(aes(col = threshold), size = 1, alpha = 0.9) +
  ggtitle("Liposarcoma vs. Normal Fat") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line()) +
  scale_colour_manual(values = c("red3", "grey", "blue", "black"),
                      breaks = c("up_sig", "not_sig", "down_sig", "other"),
                      labels = c("Up", "N.S.", "Down", "Other")) +
  geom_text_repel(data = filter(lfc_vol_df, padj < 0.1 |
                                   gene_symbol %in% c("CDK4", "MDM2", "PPARG", "CEBPA") |
                                   log2FoldChange > 2),
                  aes(label = gene_symbol), size = 2) +
  labs(col = "Significance\np.adj < 0.1")
lfc_vol_plot
```


Liposarcoma vs. Normal Fat
−log10(pvalue)
log2FoldChange
Significance
p.adj < 0.1
Up
N.S.
Down
VPS33B
SEZ6L2
EEF1AKMT3
IGLL5
TMEM132B
OLFML3
MFAP4
MARCHF9
METTL1
CYP27B1
MYO7A
AVIL
C19orf38
ASIC1
BEST3
FN1
FRAS1
GLI2
IL4I1
INHBA
XIRP1
CNKSR3
VWCE
PM20D1
RAC2
SYT1
CDK4
DUSP2
ADAMTS17
MTCL1
B4GALNT1
KRR1
CCT2
COL7A1
XIRP2
PTN
ACAN
PUS7
RAB3IP
PPARG
MYRFL
CEBPA
SLC11A2