Introduction

Type 1 diabetes (T1D) is an autoimmune disease that results in the destruction of insulin producing pancreatic B-cells, with tyrosine kinase 2 (TYK2) being a gene that plays a critical role in this autoimmunity. With a loss of function of the TYK2 gene in B-cells, it was found that the progression of T1D was halted and losing TYK2 can protect against type 1 diabetes. This study was performed to observe the role of the TYK2 gene in B-cell development while confirming that inhibiting this gene is successful at stopping type 1 diabetes progression. The authors used deep RNA sequencing to observe the expression patterns of TYK2 and other candidate genes. Also, RNA sequencing was used to compare the expression of genes in both the wild type (WT) and TYK2 knockout (KO) groups to observe any differences with the absence of TYK2.

Methods

We obtained paired-end sequencing read data for three control groups (WT) at different developmental stages and three experimental groups (KO) at different developmental stages from online resources. The read data was split across two files (R1 and R2). We also obtained a primary assembly and annotation of the human genome from online resources. We assessed the quality of the raw read data with defualt paramters of FastQC v0.12.1 [Andrews, 2010], which allowed us to check sequence quality and GC content. Then, all of the FastQC quality control results were combined and summarized using MultiQC v1.25 [Ewels et al., 2016], overwriting previous reports (-f). Next, we indexed and processed the reference genome and the annotation file using STAR v2.7.11b [Dobin et al., 2013], where we used default parameters. We then aligned the sequencing reads to the indexed reference genome using STAR [Dobin et al., 2013], where we decompressed the input reads (–readFilesCommand zcat) and produced unsorted BAMs (–outSAMtype BAM Unsorted). We also redirected the standard error to the log file with STAR (2> ${sample_id}.Log.final.out). We then parsed through the gene annotation and mapped the Gene IDs to the names of the genes using Biopython v1.84 [Cock et al., 2009] and a python script. Then, we counted genes in each sample group to quantify gene-level expression from the BAM alignments using VERSE v0.1.5 [Zhu et al., 2016], specifying a single-end counting mode for exons (-S). Finally, we merged all of the gene counts from each sample group (6 total) into a single matrix using the default parameters in pandas v2.2.3 [The pandas development team, 2025].

Quality Control Evaluation

The number of reads for all samples ranged from 85 million reads to 118 million reads. The alignment rate for all groups, experimental and control, were in the range of 95-98%. There were overrepresented sequences present, however they were found in such low numbers that they likely did not impact the results. There was high duplication found in the FastQC analysis, but this could likely be attributed to the depth of the reads. In terms of differential expression analysis, 100 million reads is very deep, which may have caused issues with duplication. However, due to a high number of reads, a high alignment rate, and a low percentage of overrepresented sequence, we can determine that this was a high quality experiment that is suitable for downstream analysis.

```
library(DESeq2)
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

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

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, 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, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
## 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
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ lubridate::%within%() masks IRanges::%within%()
## ✖ dplyr::collapse()     masks IRanges::collapse()
## ✖ dplyr::combine()      masks Biobase::combine(), BiocGenerics::combine()
## ✖ dplyr::count()        masks matrixStats::count()
## ✖ dplyr::desc()         masks IRanges::desc()
## ✖ tidyr::expand()       masks S4Vectors::expand()
## ✖ dplyr::filter()       masks stats::filter()
## ✖ dplyr::first()        masks S4Vectors::first()
## ✖ dplyr::lag()          masks stats::lag()
## ✖ ggplot2::Position()   masks BiocGenerics::Position(), base::Position()
## ✖ purrr::reduce()       masks GenomicRanges::reduce(), IRanges::reduce()
## ✖ dplyr::rename()       masks S4Vectors::rename()
## ✖ lubridate::second()   masks S4Vectors::second()
## ✖ lubridate::second<-() masks S4Vectors::second<-()
## ✖ dplyr::slice()        masks IRanges::slice()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library('fgsea')
library('ggrepel')
```

```
coldata <- tibble(samples = c('control_rep2','exp_rep2','exp_rep1','control_rep1','exp_r
ep3','control_rep3'), condition = c('control', 'exp', 'exp', 'control', 'exp', 'contro
l'))

cts <- as.matrix(read.csv('results/merged_counts.csv', row.names = 'gene'))
```

Filtering the Counts Matrix

In order to filter the counts matrix to make the data more manageable, genes were removed that did not contribute to the understanding of diffrential gene expression. More specifically, genes that showed a count of 0 for each experimental and control group were removed. If any of the groups contained at least one count of a gene, then that gene was kept in the counts matrix. The original counts matrix had 63,241 genes present. After filtering genes that had 0 counts, there were 39,893 genes remaining meaning that 23,348 genes were removed as seen in the table below.

```
keep <- rowSums(cts) > 0
cts_filter <- cts[keep, ]

before <- nrow(cts)
after <- nrow(cts_filter)
removed <- before - after

summary <- data.frame(
  Status = c("Before Filtering", "After Filtering", "Number of Genes Removed"),
  Genes = c(before, after, removed)
)

print(summary)
```

```
##                    Status Genes
## 1        Before Filtering 63241
## 2         After Filtering 39893
## 3 Number of Genes Removed 23348
```

```
dds <- DESeqDataSetFromMatrix(countData = cts_filter, colData = coldata, design = ~ cond
ition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds <- DESeq(dds)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
res <- results(dds)
resOrdered <- res[order(res$pvalue),]
resOrdered
```

```
## log2 fold change (MLE): condition exp vs control
## Wald test p-value: condition exp vs control
## DataFrame with 39893 rows and 6 columns
##                       baseMean log2FoldChange     lfcSE         stat
##                      <numeric>      <numeric> <numeric>    <numeric>
## ENSG00000129824.16   10669.168       -8.77467  0.143343     -61.2146
## ENSG00000289575.1      730.127        3.69701  0.126024      29.3357
## ENSG00000108439.11     604.080       -4.15783  0.142588     -29.1597
## ENSG00000253846.3      495.791        4.47982  0.199409      22.4654
## ENSG00000250616.4      524.454       -2.42114  0.117621     -20.5842
## ...                        ...            ...       ...          ...
## ENSG00000255883.1     11.16254   -1.12773e-04 0.5850727 -1.92751e-04
## ENSG00000163938.17 11425.03348    1.18939e-05 0.0749098  1.58776e-04
## ENSG00000159685.11  1850.89429   -2.24664e-05 0.1489736 -1.50808e-04
## ENSG00000229835.2      5.88299   -6.25972e-05 0.8415699 -7.43814e-05
## ENSG00000232872.2     52.69026   -3.55702e-07 0.2985495 -1.19143e-06
##                          pvalue         padj
##                       <numeric>    <numeric>
## ENSG00000129824.16  0.00000e+00  0.00000e+00
## ENSG00000289575.1  3.64166e-189 3.74308e-185
## ENSG00000108439.11 6.28618e-187 4.30750e-183
## ENSG00000253846.3  9.03968e-112 4.64572e-108
## ENSG00000250616.4   3.80178e-94  1.56306e-90
## ...                         ...          ...
## ENSG00000255883.1      0.999846     0.999928
## ENSG00000163938.17     0.999873     0.999928
## ENSG00000159685.11     0.999880     0.999928
## ENSG00000229835.2      0.999941           NA
## ENSG00000232872.2      0.999999     0.999999
```

```
results <- resOrdered %>% as_tibble(rownames = 'gene')
map <- read_delim('results/id2name.txt', col_names = c('id', 'symbol'))
```

```
## Rows: 63241 Columns: 2
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: "\t"
## chr (2): id, symbol
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
results <- results %>% left_join(map, by = join_by(gene == id))
results
```

```
## # A tibble: 39,893 × 8
##    gene           baseMean log2FoldChange lfcSE  stat    pvalue      padj symbol
##    <chr>             <dbl>          <dbl> <dbl> <dbl>     <dbl>     <dbl> <chr>
##  1 ENSG000001298…   10669.          -8.77 0.143 -61.2 0         0         RPS4Y1
##  2 ENSG000002895…     730.           3.70 0.126  29.3 3.64e-189 3.74e-185 ENSG0…
##  3 ENSG000001084…     604.          -4.16 0.143 -29.2 6.29e-187 4.31e-183 PNPO
##  4 ENSG000002538…     496.           4.48 0.199  22.5 9.04e-112 4.65e-108 PCDHG…
##  5 ENSG000002506…     524.          -2.42 0.118 -20.6 3.80e- 94 1.56e- 90 YPEL3…
##  6 ENSG000002511…    1300.          -1.72 0.100 -17.1 7.22e- 66 2.47e- 62 LINC0…
##  7 ENSG000001732…     633.           4.68 0.274  17.1 1.53e- 65 4.49e- 62 SLC2A…
##  8 ENSG000002863…     906.          -1.72 0.116 -14.8 2.73e- 49 7.01e- 46 ENSG0…
##  9 ENSG000002894…     383.          -2.13 0.145 -14.7 8.98e- 49 2.05e- 45 ENSG0…
## 10 ENSG000002829…     121.           3.76 0.267  14.1 3.89e- 45 8.00e- 42 ENSG0…
## # ℹ 39,883 more rows
```

```
top10 <- results %>%
  arrange(padj) %>%
  select(gene, symbol, baseMean, log2FoldChange, lfcSE, stat, pvalue, padj) %>%
  slice_head(n = 10)

top10
```

```
## # A tibble: 10 × 8
##    gene           symbol baseMean log2FoldChange lfcSE  stat    pvalue      padj
##    <chr>          <chr>     <dbl>          <dbl> <dbl> <dbl>     <dbl>     <dbl>
##  1 ENSG000001298… RPS4Y1   10669.          -8.77 0.143 -61.2 0         0
##  2 ENSG000002895… ENSG0…     730.           3.70 0.126  29.3 3.64e-189 3.74e-185
##  3 ENSG000001084… PNPO       604.          -4.16 0.143 -29.2 6.29e-187 4.31e-183
##  4 ENSG000002538… PCDHG…     496.           4.48 0.199  22.5 9.04e-112 4.65e-108
##  5 ENSG000002506… YPEL3…     524.          -2.42 0.118 -20.6 3.80e- 94 1.56e- 90
##  6 ENSG000002511… LINC0…    1300.          -1.72 0.100 -17.1 7.22e- 66 2.47e- 62
##  7 ENSG000001732… SLC2A…     633.           4.68 0.274  17.1 1.53e- 65 4.49e- 62
##  8 ENSG000002863… ENSG0…     906.          -1.72 0.116 -14.8 2.73e- 49 7.01e- 46
##  9 ENSG000002894… ENSG0…     383.          -2.13 0.145 -14.7 8.98e- 49 2.05e- 45
## 10 ENSG000002829… ENSG0…     121.           3.76 0.267  14.1 3.89e- 45 8.00e- 42
```

```
padj_threshold <- 0.05

sig_genes <- results %>%
  filter(padj < padj_threshold)

num_sig <- nrow(sig_genes)
num_sig
```

```
## [1] 1208
```

```
library(enrichR)
```

```
## Welcome to enrichR
## Checking connection ...
```

```
## Enrichr ... Connection is Live!
## FlyEnrichr ... Connection is Live!
## WormEnrichr ... Connection is Live!
## YeastEnrichr ... Connection is Live!
## FishEnrichr ... Connection is Live!
## OxEnrichr ... Connection is Live!
```

```
dbs <- c("GO_Biological_Process_2023")
sig_gene_symbols <- sig_genes$symbol %>% na.omit() %>% unique()
enrichr_results <- enrichr(sig_gene_symbols, dbs)
```

```
## Uploading data to Enrichr... Done.
##   Querying GO_Biological_Process_2023... Done.
## Parsing results... Done.
```

```
head(enrichr_results[["GO_Biological_Process_2023"]])
```

```
##                                                  Term Overlap      P.value
## 1 Positive Regulation Of Blood Coagulation (GO:0030194)   11/21 7.504999e-09
## 2      Negative Regulation Of Fibrinolysis (GO:0051918)    8/11 2.425751e-08
## 3               Nervous System Development (GO:0007399)  57/433 2.435619e-08
## 4                       Kidney Development (GO:0001822)   19/71 2.566609e-08
## 5             Regulation Of Cell Migration (GO:0030334)  56/434 6.461344e-08
## 6 Negative Regulation Of Blood Coagulation (GO:0030195)   12/30 6.983406e-08
##   Adjusted.P.value Old.P.value Old.Adjusted.P.value Odds.Ratio Combined.Score
## 1     2.282999e-05           0                   0  17.259983      322.89453
## 2     2.282999e-05           0                   0  41.753333      732.12548
## 3     2.282999e-05           0                   0   2.425532       42.52074
## 4     2.282999e-05           0                   0   5.758879      100.65424
## 5     3.590971e-05           0                   0   2.368056       39.20279
## 6     3.590971e-05           0                   0  10.464883      172.43138
##
Genes
## 1
THBD;VTN;PLAU;SERPINE1;SERPINF2;CNTN1;PLAT;EMILIN1;HPSE;F2;THBS1
## 2
THBD;VTN;PLAU;SERPINE1;SERPINF2;PLAT;F2;THBS1
## 3 CHRM2;ROBO2;TENM1;PLXND1;MYT1L;VLDLR;LDB2;CELSR1;SHH;NRCAM;CHST8;SRRM4;TMOD2;PAX6;O
LFM1;ZEB2;ALDH1A2;TYRO3;DCX;TAGLN3;RAPGEF5;DSCAML1;NUMBL;PLPPR1;C3ORF70;DLX5;NRXN1;CRMP
1;EFNA5;NEUROD1;UGT8;ACVR1C;FLRT3;ERBB4;SPOCK2;PDGFC;SLITRK6;SPOCK1;NPTX1;GPM6B;EGR2;JAG
1;ZBTB16;PHOX2B;NFASC;FGF14;DLG4;FGF19;MAB21L2;PCDHB16;NEURL1;PMP22;PCDHB5;SERPINI1;MELT
F;CBLN1;NEUROG3
## 4
ROBO2;TFAP2A;TFAP2B;TGFB2;ZBTB16;GREB1L;TCF21;LRP4;GDF6;SULF1;SULF2;PYGO1;BMP4;SHH;SALL
1;WT1;ITGA8;CTSH;REN
## 5                              RET;PTPRT;PTPRU;IFITM1;CSF1;PLXND1;CITED2;SERPINE1;TN
C;LDB2;CLDN1;AMOT;DOCK10;SHH;DACH1;FGF9;PLAU;DPYSL3;KDR;CYP1B1;CTSH;EMILIN1;PHACTR1;PLXN
C1;LIMCH1;HGF;ADRA2A;CLDN4;MMP14;SFRP1;SFRP2;CEACAM6;CDH13;SGK1;EPHA2;SEMA3C;PDGFA;THY1;
AIF1;THBS1;ERBB4;PDGFD;PDGFC;ZNF703;LMNA;CTNNA2;CGA;JAG1;SULF1;PODN;IGSF10;DAB2;FGF19;TR
IP6;CNTN1;SPRY2
## 6
FGB;THBD;VTN;C1QTNF1;PLAU;NOS3;SERPINE1;SERPINF2;PDGFA;PLAT;APOE;F2
```

```
rnks <- results %>% arrange(desc(log2FoldChange)) %>% drop_na(log2FoldChange) %>% distin
ct(symbol, .keep_all = TRUE) %>% pull(log2FoldChange, symbol)
c2 <- fgsea::gmtPathways('c2.cp.v2025.1.Hs.symbols.gmt')

fgsea_results <- fgsea(c2, rnks, minSize = 15, maxSize = 500)
```

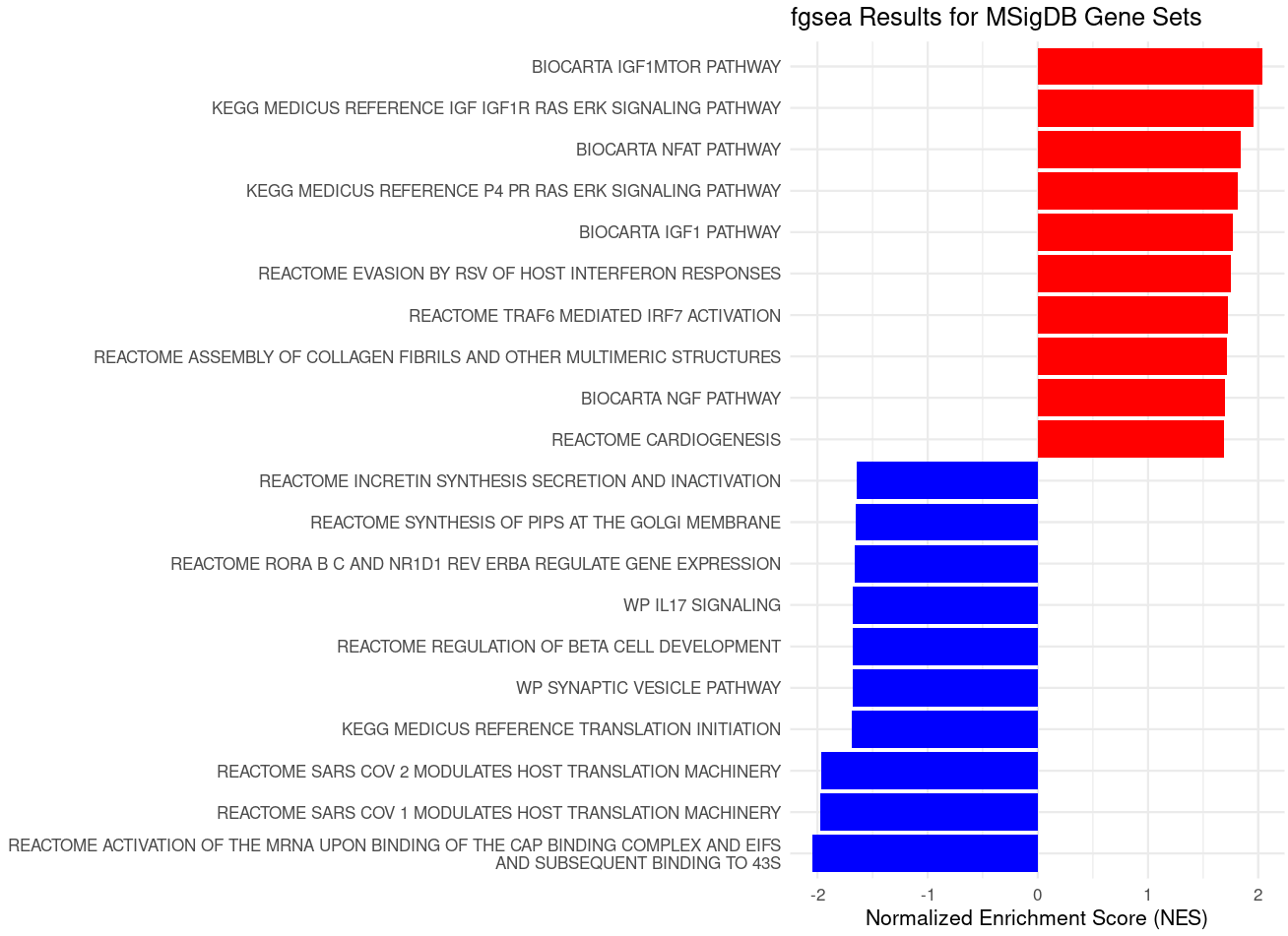
```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : Th
ere are ties in the preranked stats (25.48% of the list).
## The order of those tied genes will be arbitrary, which may produce unexpected result
s.
```

```
fgsea_results <- fgsea_results %>% as_tibble()
```

```
top_pos <- fgsea_results %>% slice_max(NES, n = 10) %>% pull(pathway)
top_neg <- fgsea_results %>% slice_min(NES, n = 10) %>% pull(pathway)

subset <- fgsea_results %>% filter(pathway %in% c(top_pos, top_neg)) %>% mutate(pathway
= factor(pathway)) %>% mutate(plot_name = str_replace_all(pathway, '_', ' '))

subset %>%
  mutate(plot_name = forcats::fct_reorder(factor(plot_name), NES)) %>%
  ggplot() +
  geom_bar(aes(x = plot_name, y = NES, fill = NES > 0), stat = 'identity', show.legend =
FALSE) +
  scale_fill_manual(values = c('TRUE' = 'red', 'FALSE' = 'blue')) +
  theme_minimal(base_size = 8) +
  ggtitle('fgsea Results for MSigDB Gene Sets') +
  ylab('Normalized Enrichment Score (NES)') +
  xlab('') +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 80)) +
  coord_flip()
```



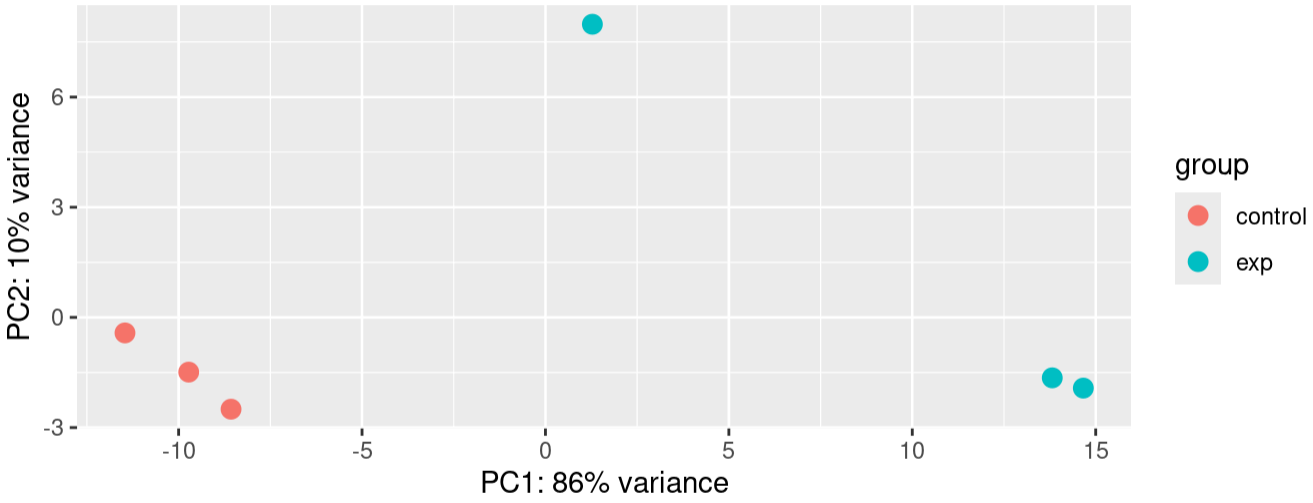
Differential Expression Analysis

With a padj threshold of 0.05, there were 1208 significant genes. With ENRICHR, the top significant genes contained Gene Ontology terms like ‘Positive Regulation Of Blood Coagulation’, ‘Nervous System Development’, and ‘Kidney Development’. This indicates that the TYK2 knockout condition alters genes involved in blood coagulation and tissue development which may have impacts on pathways linked to T1D. Overall, this indicates

that the loss of TYK2 may influence two processes that are associated with T1D, further showcasing the link between TYK2 knockout and T1D being halted. The fgsea analysis shows that knocking out TYK2 alters key signaling and immune pathways related to T1D like IGF1MTOR, NFAT, and ERK. These pathways are associated with immune activation and cell growth, suggesting that these processes are upregulated with TYK2 present. On the other hand, TYK2 knockout samples show enrichment of pathways related to translation and transcriptional regulation. Overall, the data suggests that the loss of TYK2 in the experimental samples dampens immune activation and inflammatory signaling by IGF1MTOR, NFAT, and ERK, which may help protect B-cells from an autoimmune attack with T1D. One similarity between the fgsea analysis and the ENRICHR analysis is that they both indicate that TYK2 knockouts in the experimental samples impact immune-related and regulatory pathways linked to T1D. This further supports that TYK2 knockout may be helpful when looking to halt T1D progression. On the other hand, ENRICHR highlights more broad biological processes while fgsea focuses on more specific signaling pathways. Perhaps using DAVID over ENRICHR would lead to more similar results in terms of specificity between the two analyses. Overall, both analyses indicate that TYK2 loss downregulates immune activation, which is relevant to the progression of T1D.

```
vsd <- vst(dds, blind = TRUE)
plotPCA(vsd, intgroup = c("condition"))
```

```
## using ntop=500 top features by variance
```

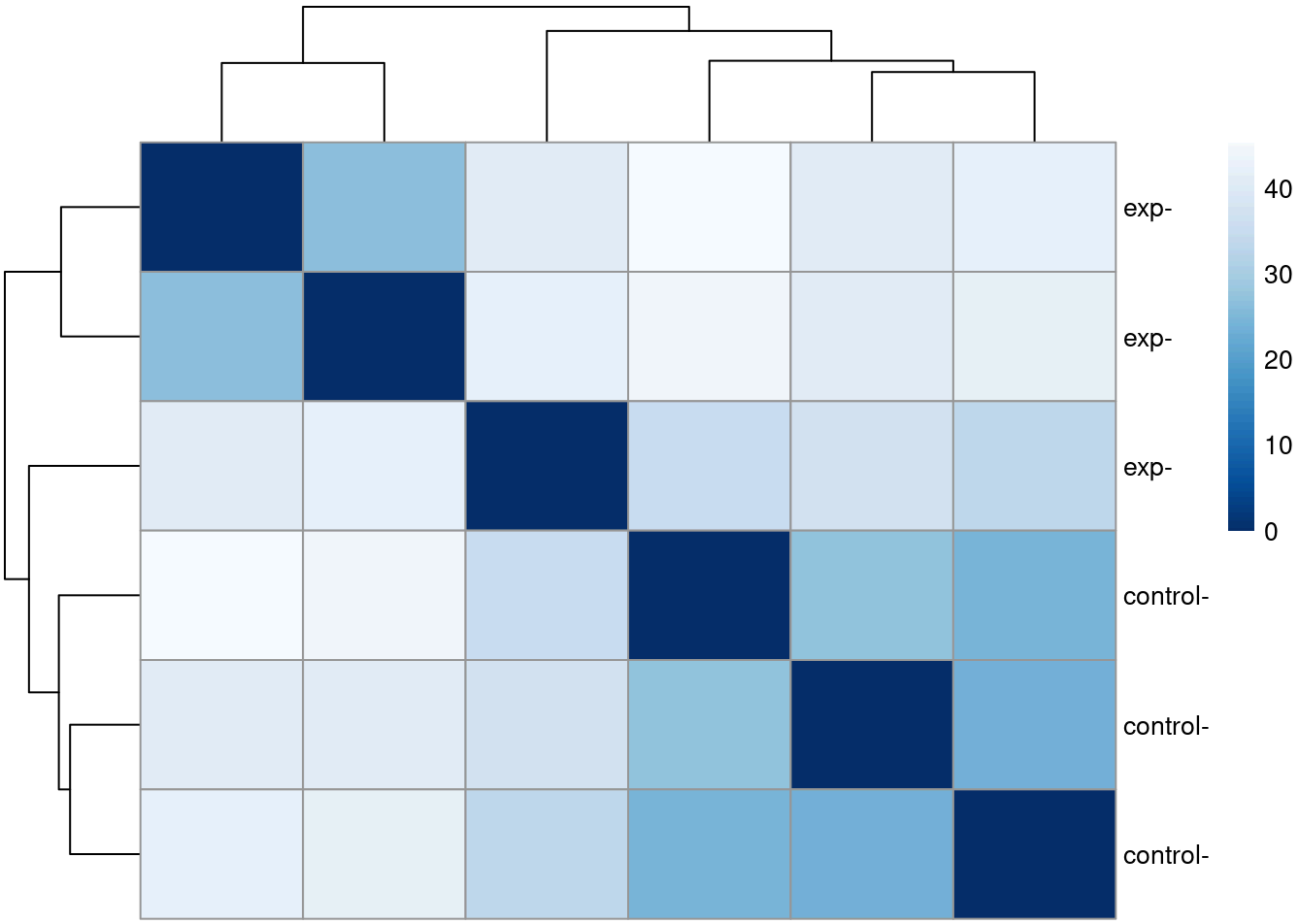


```
library("RColorBrewer")
library('pheatmap')

sampleDists <- dist(t(assay(vsd)))
sampleDistsMatrix <- as.matrix(sampleDists)

rownames(sampleDistsMatrix) <- paste(vsd$condition, vsd$type, sep = "-")
colnames(sampleDistsMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)

pheatmap(sampleDistsMatrix, clustering_distance_rows = sampleDists, clustering_distance_
cols = sampleDists, col = colors)
```



RNAseq Quality Control

PCA is an exploratory data analysis technique that is used as a diagnostic to see if the samples separate and group by condition. In this experiment, we can see that the experimental samples and the control samples seem to group together, indicating that the knockout gene may be responsible for the change in gene expression that we see and is driving this variance. There is also one outlier in the experimental group, but since this sample is still separated from the control group, it is not currently an issue for this analysis. Overall, since our samples are in two separate groups, we would expect to see a clear difference between these groups in the differential expression results. The distance matrix indicates how similar and different two different experimental samples are to one another. This plot is also used as a diagnostic where the control samples are expected to be more similar to one another while the experimental samples are expected to be more similar to each other. If there is a biological difference between the two groups, then we would expect for them to be very different from one

another in the distance matrix. This plot is indicative that this expected relationship is true. The experimental samples are more closely related and grouped together separately from the control samples, which are also grouped together and shown to be more similar. Overall, both of these plots are indicating that the results we observe in our differential expression are due to the knockout condition.

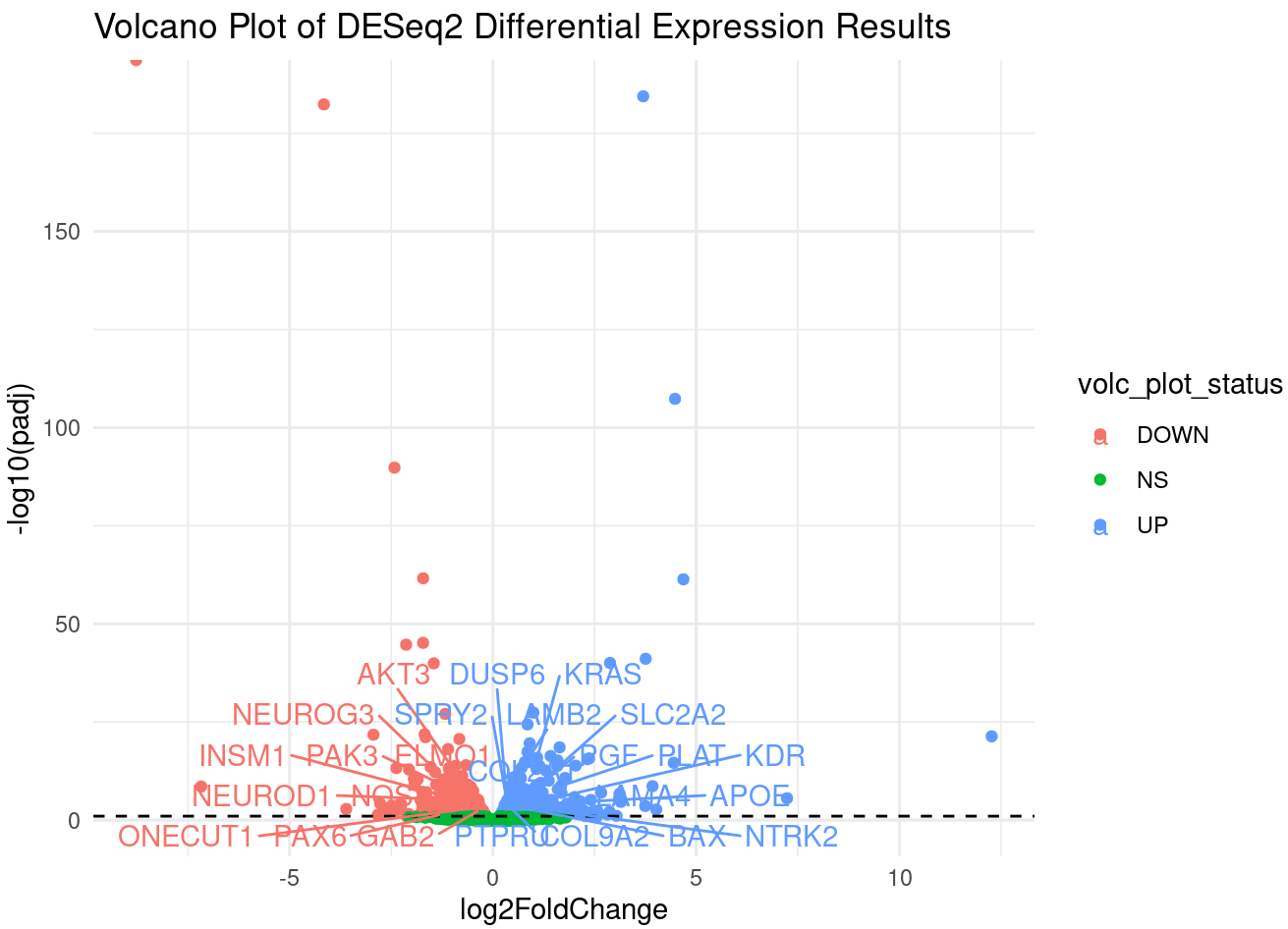
```
selected <- c('ELMO1', 'PAK3', 'INSM1', 'NEUROG3', 'GAB2', 'NOS3', 'PAX6', 'NEUROD1', 'O
NECUT1', 'KRAS', 'LAMB2', 'SPP', 'DUSP6', 'SPRY2', 'PLAT', 'AKT3', 'SLC2A2', 'BAX', 'COL
2A1', 'APOE', 'LAMA4', 'KDR', 'PGF', 'PTPRU', 'COL9A2', 'NTRK2')

results <- results %>% mutate(label = if_else(symbol %in% selected, TRUE, FALSE))
```

```
labeled_results <- results %>% mutate(volc_plot_status = case_when(
  log2FoldChange > 0 & padj < 0.15 ~ 'UP',
  log2FoldChange < 0 & padj < 0.15 ~ 'DOWN',
  TRUE ~ 'NS'))
  #label = padj < 0.05 & abs(log2FoldChange) > 1)

labeled_results %>%
  ggplot(mapping = aes(x = log2FoldChange, y = -log10(padj), color = volc_plot_status))
+
  geom_point() +
  geom_hline(yintercept = -log10(0.1), linetype = "dashed") +
  geom_text_repel(data = labeled_results %>% filter(label == TRUE), aes(label = symbol),
max.overlaps = 100) +
  theme_minimal() +
  ggtitle('Volcano Plot of DESeq2 Differential Expression Results')
```

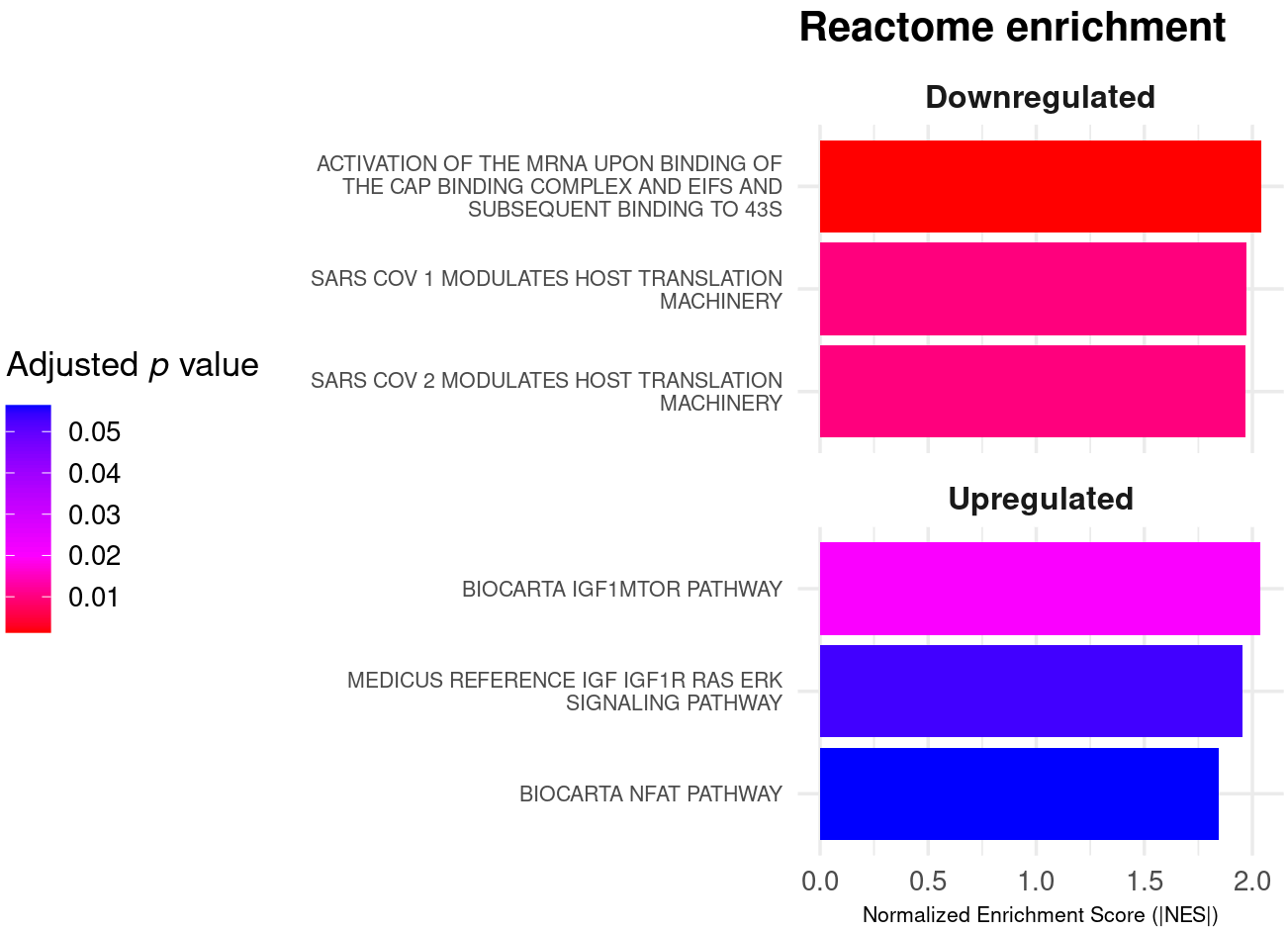
```
## Warning: Removed 19336 rows containing missing values or values outside the scale ran
ge
## (`geom_point()`).
```


Volcano Plot of DESeq2 Differential Expression Results
-log10(padj)
150
100
50
0
-5
0
5
10
log2FoldChange
AKT3
DUSP6
KRAS
NEUROG3
SPRY2
LAMB2
SLC2A2
INSM1
PAK3
FLNC
PGF
PLAT
KDR
NEUROD1
APOE
ONECUT1
PAX6
GAB2
PTPRU
COL9A2
BAX
NTRK2
volc_plot_status
DOWN
NS
UP

```
library(dplyr)
library(ggplot2)
library(fgsea)
library(stringr)
library(tibble)

plot_df <- fgsea_results %>%
  as_tibble() %>%
  arrange(padj) %>%
  mutate(Direction = ifelse(NES > 0, "Upregulated", "Downregulated")) %>%
  group_by(Direction) %>%
  slice_head(n = 3) %>%
  ungroup() %>%
  mutate(
    pathway = gsub("REACTOME_|KEGG_|WP_", "", pathway),
    pathway = gsub("_", " ", pathway),
    pathway_short = str_wrap(pathway, width = 40),
    pathway_short = factor(pathway_short, levels = rev(unique(pathway_short)))
  )

ggplot(plot_df, aes(x = abs(NES), y = pathway_short, fill = padj)) +
  geom_col() +
  facet_wrap(~Direction, scales = "free_y", ncol = 1) +
  scale_fill_gradientn(
    colors = c("red", "magenta", "blue"),
    values = scales::rescale(c(0.01, 0.02, 0.04)),
    name = expression("Adjusted " * italic(p) * " value")
  ) +
  labs(
    x = "Normalized Enrichment Score (|NES|)",
    y = NULL,
    title = "Reactome enrichment"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    axis.text.y = element_text(size = 8),
    axis.title.x = element_text(size = 8),
    plot.title = element_text(face = "bold"),
    legend.position = "left",
    strip.text = element_text(size = 12, face = "bold")
  )
```



Replicate Figure 3C and 3F

With a significance threshold of 0.01, the paper found a total of 731 significant genes, 319 of which were upregulated and 412 of which were downregulated. In comparison, the above analysis found a total of 698 significant genes, with the number of those being upregulated and downregulated unknown. Given the similar total number of significant genes, it is possible to assume that both of the analyses were similar and provided similar results. The enrichment results from the provided paper showcased that there was downregulation of gene sets crucial to proper endocrine development. For example, there was downregulation of gene sets related to “Regulation of B-cell Development” and “Gene Expression in B-cells” in the TYK2 knockout samples. Also, the samples from the paper displayed lower expression of gene sets critical to antigen processing and presentation. This is important because it indicates that TYK2 loss is protective for the immune system in cells with T1D. One similarity between the paper’s analysis and the above analysis is that they both indicate that TYK2 loss downregulates immune activation and dampen inflammatory responses related to T1D progression. Some differences and discrepancies between the two analyses may appear when comparing the two experiments. In comparison to the above analysis, the paper completes a more thorough analysis where they filter based on multiple time points in cell development, while this analysis is focusing only on S5 time points. The paper may have used a different version of the human reference genome than the above analysis, leading to discrepancies in the data. Aligning reads to a larger genome does not always happen the exact same way, and perhaps the paper had some varying alignments that we did not observe. Also, all of the tools used in the pipeline could have been different versions, leading to varying results. Overall, while there are many discrepancies and differences that may have occurred, we were generally able to replicate this experiment with similar resulting pathways that the experiment in the paper found.

Literature Cited

Andrews S. (2010). FastQC: a quality control tool for high throughput sequence data. Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc (http://www.bioinformatics.babraham.ac.uk/projects/fastqc)

Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics. 2016 Oct 1;32(19):3047-8. doi: 10.1093/bioinformatics/btw354. Epub 2016 Jun 16. PMID: 27312411; PMCID: PMC5039924.

Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013 Jan 1;29(1):15-21. doi: 10.1093/bioinformatics/bts635. Epub 2012 Oct 25. PMID: 23104886; PMCID: PMC3530905.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczyński, Michiel J. L. de Hoon: Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25 (11), 1422–1423 (2009). https://doi.org/10.1093/bioinformatics/btp163 (https://doi.org/10.1093/bioinformatics/btp163)

Zhu, Qin & Fisher, Stephen & Shallcross, Jamie & Kim, Junhyong. (2016). VERSE: a versatile and efficient RNA-Seq read counting tool. 10.1101/053306.

The pandas development team. (2025). pandas-dev/pandas: Pandas (v2.2.3). Zenodo. https://doi.org/10.5281/zenodo.17229934 (https://doi.org/10.5281/zenodo.17229934)