homework_05

Esteban Jorquera; Marlon Aldair; Diego Ramírez

2022-03-20

Github link: https://github.com/EstebanJorquera/UNAM_BIOinfoII.git

```
# Note installing packages through knitter is not the best idea,
# the following code will likely fail if the package is not already installed, with a missing mirror er
# for this reason there's an included R script just for that
# anyway this meant to be run on a cluster rather than locally, so happy copy pasting :)!
# Installs required packages if not already installed (avoids re installing)
if (!requireNamespace("BiocManager", quietly = TRUE))     install.packages("BiocManager")

if (!requireNamespace("tidyverse", quietly = TRUE))       install.packages("tidyverse")
if (!requireNamespace("dplyr", quietly = TRUE))           install.packages("dplyr")
if (!requireNamespace("tidyr", quietly = TRUE))           install.packages("tidyr")
if (!requireNamespace("ggplot2", quietly = TRUE))         install.packages("ggplot2")

# Libraries
library(tidyverse)
library(dplyr)
library(tidyr)
library(ggplot2)

BiocManager::install(version = "3.13")
packages = c("DESeq2", "tximport")
BiocManager::install(packages)
```

```
#!/bin/bash
# Use current working directory
#$ -cwd
#
# Join stdout and stderr
#$ -j y
#
# Run job through bash shell
#$ -S /bin/bash
#
# You can edit the script since this line
#
# Your job name
#$ -N EJ_multiqc

# Send an email after the job has finished
#$ -m e
```

```
#$ -M eijorquera@uc.cl
#
# Line required if modules are to be used, source modules environment
. /etc/profile.d/modules.sh
#
# quality control and trimming of reads,
# loads fastqc, trimmomatic and multiqc module, also downloads Truseq adapters
# fastqc is used to do qc analysis of the ChIP-seq reads
# multiqc generates a report for the non-trimmed reads
# wget downloads adapter file
# trimmomatic trimms the adapters from the reads using the adapter file as reference (ToDo check how to
# fastqc is used to do qc analysis of the trimmed ChIP-seq reads
# multiqc generates a report for the trimmed reads
(module load fastqc/0.11.3 ;
module load trimmomatic/0.33 ;
module load multiqc/1.5 ;
fastqc /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/*.fastq.gz -o /mnt/Citosina/amedina/
multiQC /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/QC1 ;
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417885
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417886
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417887
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417888
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417889
trimmomatic PE -phred33 -basein /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/SRR6417890
fastqc /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/*.fastq.gz -o /mnt/Citosina/amedina/ejorque
multiQC /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/QC2)

#!/bin/bash
# Use current working directory
#$ -cwd
#
# Join stdout and stderr
#$ -j y
#
# Run job through bash shell
#$ -S /bin/bash
#
# You can edit the script since this line
#
# Your job name
#$ -N EJ_kall_index

# Send an email after the job has finished
#$ -m e
#$ -M eijorquera@uc.cl
#
# Line required if modules are to be used, source modules environment
. /etc/profile.d/modules.sh
#
# loads kallisto module and downloads the latest reference transcriptome for homosapiens
# uses kallisto to index the reference transcriptome
(module load kallisto/0.45.0 ;
wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/latest_release/gencode.v39.transcripts.fa
kallisto index -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 /mnt
```

```
#!/bin/bash
# Use current working directory
#$ -cwd
#
# Join stdout and stderr
#$ -j y
#
# Run job through bash shell
#$ -S /bin/bash
#
# You can edit the script since this line
#
# Your job name
#$ -N EJ_kall_quant

# Send an email after the job has finished
#$ -m e
#$ -M eijorquera@uc.cl
#
# Line required if modules are to be used, source modules environment
. /etc/profile.d/modules.sh
#
# Loads kallisto module,
# executes kallisto to make reads counts using the transcriptome as a reference
(module load kallisto/0.45.0 ;
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
kallisto quant -i /mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/index_kallisto_gencode-h39 -o /
```

Run this to enter an R session in screen mode

screen -S txi qlogin R

```r
## Data loading

# gets the current work directory
getwd()
# sets the current work directory to kallisto's output folder
setwd("/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/kallisto")

# imports the required libraries
library(tximport)
library(tidyverse)


# kallisto pseudo alignment result counts
files <- file.path("/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/kallisto",list.dirs(dir(".")))
names(files) <- str_extract(files,"SRR\\d+")
files
```

```r
# ensembl transcript_id - gene_id equivalence table, permits assigning the proper gene ID to each trans
tx2gene <- read.csv("/mnt/Timina/bioinfoII/rnaseq/resources/gencode/gencode.v38.basic.pc.transcripts.ens

# ensembl transcript_id - gene name/symbol equivalence table, permits assigning the proper gene name to
tx2genename <- read.csv("/mnt/Timina/bioinfoII/rnaseq/resources/gencode/gencode.v38.basic.pc.transcripts

# tx2gene asigns transcript IDs to gene IDs and gene names for summarization
txi.kallisto <- tximport(files, type = "kallisto", tx2gene = tx2gene, ignoreAfterBar=TRUE, ignoreTxVersi
txi.kallisto.name <- tximport(files, type = "kallisto", tx2gene = tx2genename, ignoreAfterBar=TRUE, igno

# displays the columns in the dataframe
names(txi.kallisto)
# shows the top (not ordered) of the dataframe
head(txi.kallisto$counts)

# loads a csv table describing the samples used in the analysis, the table was manually made using the
samples <- read.csv("/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/data/fastq/samples.csv",stringsAs
samples <- column_to_rownames(samples, var = "sample")
# displays the sample dataframe
samples



## Differential expression analysis

# imports the required libraries
library(DESeq2)
dds <- DESeqDataSetFromTximport(txi.kallisto,
                                colData = samples,
                                design = ~ genotype)
dds

# Removes genes with low counts, by keeping only the rest
keep <- rowSums(counts(dds)) >= 6
dds <- dds[keep, ]
dds$genotype <- factor(dds$genotype, levels = c("Wild_Type","FOXP1_KD"))

# Runs DESeq for the filtered dataframe
dds <- DESeq(dds)

# saves and displays the results of DESeq to a results object; considers a 10% FDR
res <- results(dds)
res
summary(res)

# Similar as before but with an stricter 5% FDR
res.05 <- results(dds, alpha = 0.05)
table(res.05$padj < 0.05)
summary(res.05)

# Gets genes that show a large (more than twice, less than half) fold change upon FOXP1 knockdown
resLFC1 <- results(dds, lfcThreshold=1)
table(resLFC1$padj < 0.1)
```

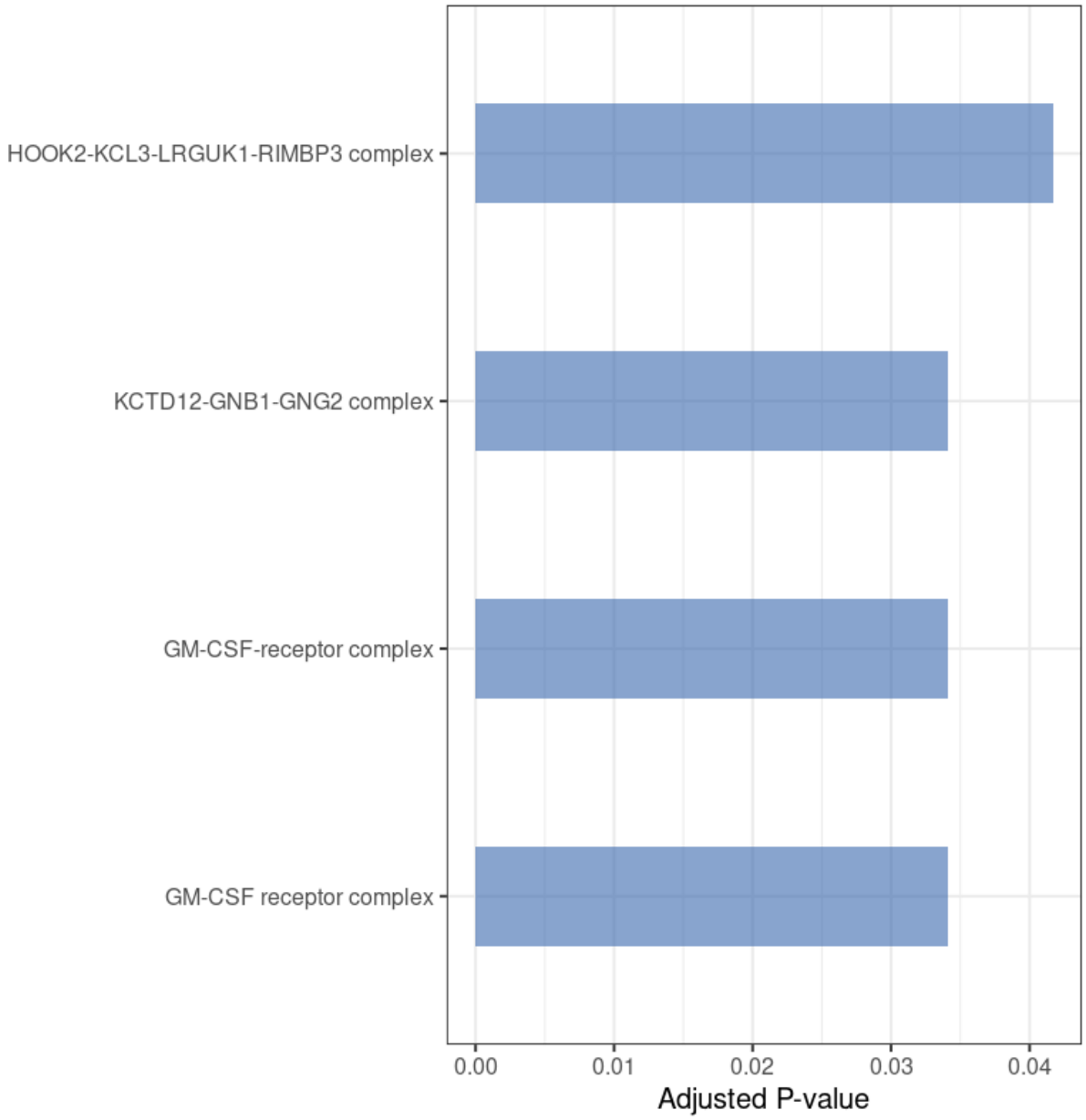
```r
summary(resLFC1)

# Filters the 10% FDR results dataframe of p-adjusted values
resSig <- subset(res, padj < 0.1)
# Shows the more downregulated genes
head(resSig[ order(resSig$log2FoldChange), ])
# Shows the more upregulated genes
head(resSig[ order(resSig$log2FoldChange, decreasing = TRUE), ])


## MA plotting
#
# BiocManager::install("apeglm") ### won't run, can't install the package
# library(apeglm) # good for shrinking the noisy LFC estimates while giving low bias LFC estimates for
#
# resultsNames(dds)
# resLFC <- lfcShrink(dds, coef="condition_Verafinib_vs_untreated", type="apeglm")
# resLFC
#
# # output folder for generated plots
# outdir = "/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/plots/"
# png(file = paste0(outdir,"maplot01-res-noshrink.png"),
#     width = 800, height = 800) # guardar el plot en formato png
# plotMA(res, ylim = c(-5, 5)) # funcion de DESeq
# dev.off()
# # resultados genes significativos
# png(file = paste0(outdir,"maplot01-resSig-noshrink.png"),
#     width = 800, height = 800) # guardar el plot en formato png
# plotMA(resSig, ylim = c(-5, 5)) # funcion de DESeq
# dev.off()
# # lfcShrink results
# png(file = paste0(outdir,"maplot01-resLFC-shrink.png"),
#     width = 800, height = 800) # guardar el plot en formato png
# plotMA(resLFC, ylim = c(-5, 5)) # funcion de DESeq
# dev.off()
# # most significative genes highlightning
# png(file = paste0(outdir,"maplot02-res-noshrink.png"),
#     width = 800, height = 800)
# plotMA(res, ylim = c(-5,5))
# # gene with the lowest p.adjusted value
# topGene <- rownames(res)[which.min(res$padj)]
# with(res[topGene, ], {
#   points(baseMean, log2FoldChange, col="dodgerblue", cex=2, lwd=2)
#   text(baseMean, log2FoldChange, topGene, pos=2, col="dodgerblue")
# })
# dev.off()



## Volcano plotting

# imports the required libraries
library(ggplot2)
```

```
# output folder for generated plots
outdir = "/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/plots/"
png(file = paste0(outdir,"volcano01-res.png"),
    width = 800, height = 800) # guardar el plot en formato png
# The basic scatter plot: x is "log2FoldChange", y is "pvalue"
ggplot(data=as.data.frame(res), aes(x=log2FoldChange, y=pvalue)) +
  geom_point() # scatter plot
dev.off()

png(file = paste0(outdir,"volcano02-res.png"),
    width = 800, height = 800) # guardar el plot en formato png
ggplot(data=as.data.frame(res), aes(x=log2FoldChange, y=-log10(pvalue))) +
  geom_point() # scatter plot
dev.off()

png(file = paste0(outdir,"volcano03-res.png"),
    width = 800, height = 800) # guardar el plot en formato png
ggplot(data=as.data.frame(res), aes(x=log2FoldChange, y=-log10(pvalue))) +
  geom_point() + # scatter plot
  theme_minimal() + # tema de fondo
  geom_vline(xintercept=c(-0.6, 0.6), col="red") + # vertical lines for log2FoldChange thresholds
  geom_hline(yintercept=-log10(0.05), col="red") + # horizontal line for the p-value threshold
  xlim(-15,15)
dev.off()

# Add a column to the data frame to specify if they are UP- or DOWN- regulated (log2FoldChange respecti
de <- as.data.frame(res)
# add a column of NAs
de$diffexpressed <- "NO"
# if log2Foldchange > 0.6 and pvalue < 0.05, set as "UP"
de$diffexpressed[de$log2FoldChange > 0.6 & de$pvalue < 0.05] <- "UP"
# if log2Foldchange < -0.6 and pvalue < 0.05, set as "DOWN"
de$diffexpressed[de$log2FoldChange < -0.6 & de$pvalue < 0.05] <- "DOWN"
png(file = paste0(outdir,"volcano04-res.png"),
    width = 800, height = 800) # guardar el plot en formato png
ggplot(data=de, aes(x=log2FoldChange, y=-log10(pvalue), col=diffexpressed)) + # cambiamos col param de
  geom_point() + theme_minimal() +
  geom_vline(xintercept=c(-0.6, 0.6), col="red") +
  geom_hline(yintercept=-log10(0.05), col="red") +
  xlim(-15,15)
dev.off()

# this library avoids label text overlapping
library(ggrepel)
# Create a new column "names" to de, that will contain the name of a subset if genes differentially exp
de$names <- NA
# filter for a subset of interesting genes
filter <- which(de$diffexpressed != "NO" & de$padj < 0.05 & (de$log2FoldChange >= 9 | de$log2FoldChange
de$names[filter] <- rownames(de)[filter]
png(file = paste0(outdir,"volcano05-res.png"),
    width = 800, height = 800)
ggplot(data=de, aes(x=log2FoldChange, y=-log10(pvalue), col=diffexpressed, label=names)) +
    geom_point() +
```

```
    scale_color_manual(values=c("blue", "black", "red")) +
    theme_minimal() +
    geom_text_repel() +
    xlim(-15,15)
dev.off()


## GO enrichment

# imports the required libraries
library(gprofiler2)


## 1st image
# subset results for genes of interest
resSig <- subset(res, padj < 0.1 & log2FoldChange > 1)
resSig <- resSig[ order(resSig$log2FoldChange, decreasing = TRUE), ]
# define gene lists
DEG <- rownames(resSig) # genes that passed the previous significance test
genes_universe <- rownames(res) # all genes with counts

# enrichment analysis, this one does not consider a background
gostres = gost(query = DEG, organism = "hsapiens", significant = T,
               correction_method = "fdr", domain_scope = "annotated", custom_bg = NULL, ordered_query =
names(gostres)
attributes(gostres$meta)
head(gostres$result)

library(forcats)
outdir = "/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/plots/"
go <- as.data.frame(gostres$result)
png(file = paste0(outdir,"gost01-resSig.png"),
    width = 800, height = 800)
# Reorder following the value of another column:
go %>%
  arrange(p_value) %>% # incremental de p.value ordering
  select(term_name,p_value) %>%  # df column selection
  dplyr::slice(1:20) %>% # top 20 rows
  mutate(go = fct_reorder(term_name, p_value)) %>%
  ggplot( aes(x=go, y=p_value)) +
    geom_bar(stat="identity", fill="#3A68AE", alpha=.6, width=.4) +
    coord_flip() +
    labs(x = "", y = "Adjusted P-value") +
    theme_bw(base_size = 20)
dev.off()


## 2nd image; note we are overwriting some of the previous objects... so re-run the whole chunk
# subset results for genes of interest
resSig <- subset(res, padj < 0.1 & log2FoldChange > 1) # subset
resSig <- resSig[ order(resSig$log2FoldChange, decreasing = TRUE), ]
# define gene lists
DEG <- rownames(resSig)
```

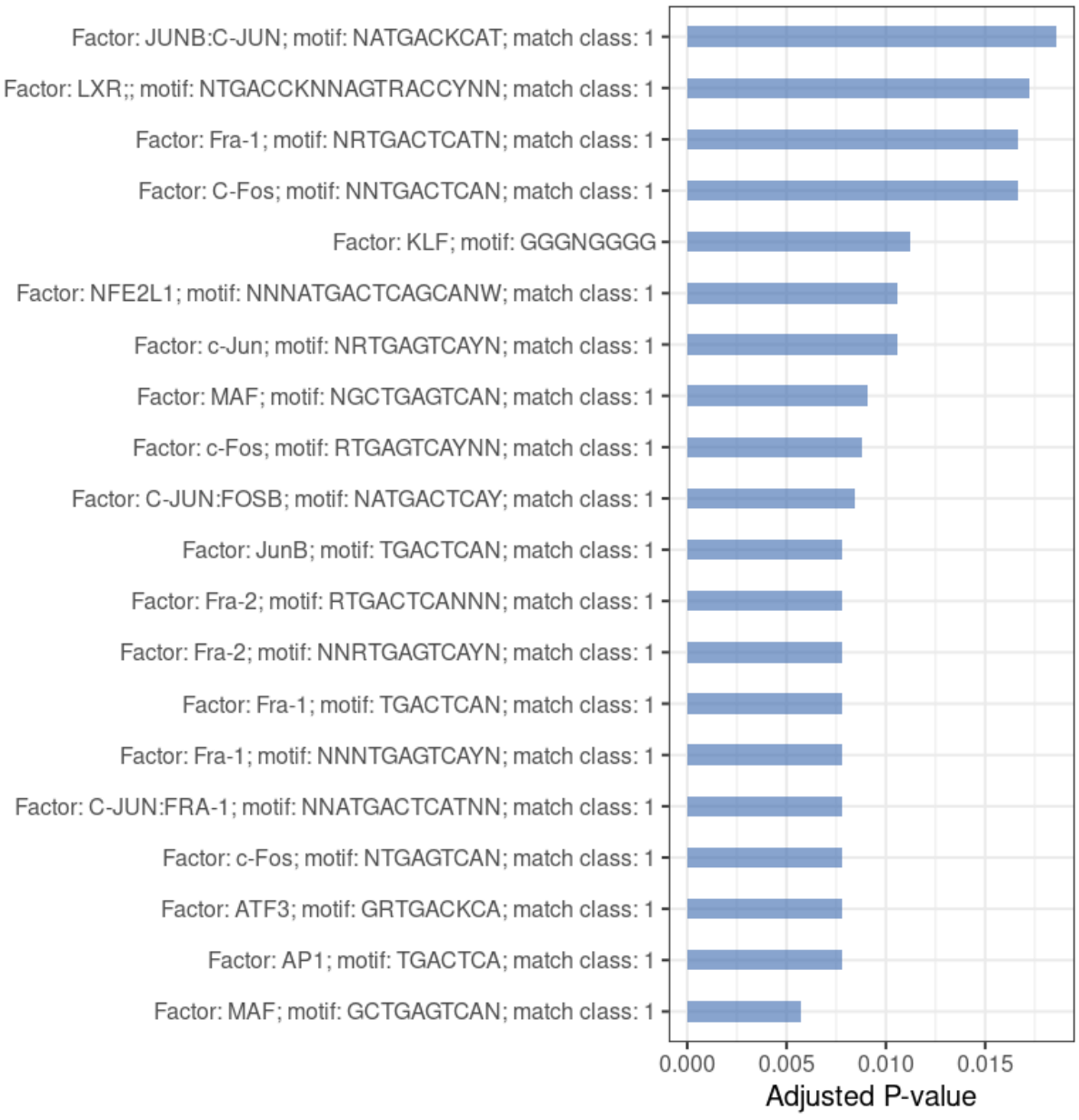
```r
genes_universe <- rownames(res)

# enrichment analysis, this one does consider a background
gostres = gost(query = DEG, organism = "hsapiens", significant = T,
               correction_method = "fdr", domain_scope = "custom",
               custom_bg = genes_universe, ordered_query = TRUE)
names(gostres)
attributes(gostres$meta)
head(gostres$result)

library(forcats)
outdir = "/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/plots/"
go <- as.data.frame(gostres$result)
png(file = paste0(outdir,"gost02-resSig.png"),
    width = 800, height = 800) # guardar el plot en formato png
# Reorder following the value of another column:
go %>%
  arrange(p_value) %>% # incremental de p.value ordering
  select(term_name,p_value) %>%  # df column selection
  dplyr::slice(1:20) %>% # top 20 rows
  mutate(go = fct_reorder(term_name, p_value)) %>%
  ggplot( aes(x=go, y=p_value)) +
    geom_bar(stat="identity", fill="#3A68AE", alpha=.6, width=.4) +
    coord_flip() +
    labs(x = "", y = "Adjusted P-value") +
    theme_bw(base_size = 20)
dev.off()


## 3rd image; note we are overwriting some of the previous objects... so re-run the whole chunk
# subset results for genes of interest
resSig <- subset(res, padj < 0.1 & log2FoldChange < -1) # subset
resSig <- resSig[ order(resSig$log2FoldChange, decreasing = FALSE), ]
# define gene lists
DEG <- rownames(resSig)
genes_universe <- rownames(res)

# enrichment analysis, this one does consider a background and it's in the reverse order (considers onl
gostres = gost(query = DEG, organism = "hsapiens", significant = T,
               correction_method = "fdr", domain_scope = "custom",
               custom_bg = genes_universe, ordered_query = TRUE)
names(gostres)
attributes(gostres$meta)
head(gostres$result)

library(forcats)
outdir = "/mnt/Citosina/amedina/ejorquera/BioInfoII/Tarea_5/out/plots/"
go <- as.data.frame(gostres$result)
png(file = paste0(outdir,"gost02-resSig.png"),
    width = 800, height = 800) # guardar el plot en formato png
# Reorder following the value of another column:
go %>%
  arrange(-p_value) %>% # incremental de p.value ordering
```

```
  select(term_name,p_value) %>%  # df column selection
  dplyr::slice(1:20) %>% # top 20 rows
  mutate(go = fct_reorder(term_name, -p_value)) %>%
  ggplot( aes(x=go, y=p_value)) +
    geom_bar(stat="identity", fill="#3A68AE", alpha=.6, width=.4) +
    coord_flip() +
    labs(x = "", y = "Adjusted P-value") +
    theme_bw(base_size = 20)
dev.off()
```



Second result generated from the gene ontology analysis of the FOXP1 differential gene expression RNA-seq data. In this image we observe the categories enriched upon FOXP1 knockdown, or in other words the genes

that are significantly upregulated (log 2 fold change > 1) when this transcription factor is absent in the A549 lung carcinoma cell line. Interestingly, we observe that gene ontology category enrichment indicates that genes related to 3 proteins complexes are being upregulated in response to the loss of FOXp1, one of these, the KCTD12-GNB1-GNG2 complex might be of special interest considering that, according to Sheng et al, 2019, one of the highlighted proteins found to be upregulated in FOXP1 KD cells was another member of the family, GNG7. Similarly, the GM-CSF complex is also upregulated, this complex acts as a cytokine receptor, which is concordant with Sheng's results, where their group found that loss of FOXP1 causes an overall increase in the expression of chemokine signalling genes which results in increased proliferation of the lung adenocarcinoma cells, suggesting that FOXP1 acts as a tumor suppressor gene.



We generated a third gene ontology analysis were we selected the genes that were significantly downregulated (log 2 fold change < -1), when the transcription factor FOXP1 is lost, or in other words, the genes that were upregulated by FOXP1 in the wild type condition, interestingly in this case we observe that the gene

ontology enriched in wild type conditions correspond to other transcription factors, this suggests to us that FOXP1 might regulate other transcription factors, and perhaps act as a master regulator in the A549 cell line.