IRF1 Sanity Check

2023-03-20

```
#load required packages.
library(biomaRt)
library(plyranges)
library(GenomicRanges)
library(genomation)
library(ondisc)
library(readr)
library(dplyr)
library(sceptre3)
library(BH)
library(varhandle)
library(kableExtra)
library(rjson)
library(ggplot2)
```

Goal

The goal of this report is to compare the genes that are found to be significantly affected by IRF1 perturbation via Seurat and SCEPTRE to CHIPseq data.

Read in Promoter Data

```
#read in papalexi data
LOCAL_SCEPTRE2_DATA_DIR <-.get_config_path("LOCAL_SCEPTRE2_DATA_DIR")
papalexi_dir <- paste0(LOCAL_SCEPTRE2_DATA_DIR, "data/papalexi/eccite_screen/")
# gene info
gene_odm_fp <- paste0(papalexi_dir, "gene/matrix.odm")
gene_metadata_fp <- paste0(papalexi_dir, "gene/metadata_qc.rds")
gene_odm <- read_odm(odm_fp = gene_odm_fp, metadata_fp = gene_metadata_fp)
```

```
#get TSS for each gene
ensembl <- useEnsembl(host = 'https://grch37.ensembl.org',biomart = 'ENSEMBL_MART_ENSEMBL',
                      dataset = "hsapiens_gene_ensembl")
A = getBM(attributes=c("hgnc_symbol", "chromosome_name", "start_position",
                       "end_position", "strand"),
          filters=c('hgnc_symbol'),
          value = gene_odm |> get_feature_ids(), mart=ensembl) |>
  dplyr::filter(chromosome_name %in% c(1:22, "X", "Y"))
```

```
#get start and end site depending on whether the strand is postive or negative
TSS_start = rep(NA,nrow(A))
TSS_end = rep(NA,nrow(A))
for(j in c(1:nrow(A))){
  #if strand positive, use [start-500,start]
  if(A$strand[j]==1){
    TSS_end[j] = A$start_position[j]
    TSS_start[j] = A$start_position[j]-500
  }else{
    #if strand negative use [end,end + 500]
    TSS_start[j] = A$end_position[j]
    TSS_end[j] = A$end_position[j]+500
  }
}
#add to A matrix
A$TSS_start = TSS_start
A$TSS_end = TSS_end
#add chr to chromosome name
A$chromosome_name = paste0("chr",A$chromosome_name)
```

```
#use A to make a promoter granges object
promoters <- GRanges(
  seqnames = A$chromosome_name,
  ranges = IRanges(start = A$TSS_start, end = A$TSS_end),
  TF = A$hgnc_symbol)
```

Read in CHIPseq Data and Join the Datasets

```
#read in chipseq data as granges object
data.dir = .get_config_path("LOCAL_SCEPTRE2_DATA_DIR")
stat1 = 'GSM935488_hg19_wgEncodeSydhTfbsK562Stat1Ifng6hStdPk.narrowPeak'
irf1 = "GSM935549_hg19_wgEncodeSydhTfbsK562Irf1Ifng6hStdPk.narrowPeak"
chipseq.dir = paste0(data.dir, "data/chipseq/", irf1)
chipseq_data = readNarrowPeak(chipseq.dir, track.line=FALSE, zero.based=TRUE)
overlap_genes = c()
for(chr in paste0('chr',c(1:23,"X","Y"))){
  bobby = subset(chipseq_data,seqnames@values == chr)
  chucky = subset(promoters,seqnames@values == chr)
  direct_effects = plyranges::join_overlap_left(chucky,bobby,
                                                 minoverlap = 1)
  overlap_genes = c(overlap_genes,direct_effects$TF[is.na(direct_effects$score)==F])
}
```

```
sceptre2_dir <- .get_config_path("LOCAL_SCEPTRE2_DATA_DIR")
# directory for hTFtarget data
htftarget_dir <- paste0(sceptre2_dir, "data/htftarget")
```

```
ref_genes = read_table(paste0(data.dir, "data/htftarget/dataset_1762.IRF1.target.txt")) |>
  pull(target_name)
```

```
# number of targets in database
length(ref_genes)
```

```
## [1] 11194
```

```
# number of targets we found
length(overlap_genes)
```

```
## [1] 327
```

```r
# number of targets overlapping
length(base::intersect(overlap_genes, ref_genes))
```

```
## [1] 268
```

```r
# number of targets unique to us
length(base::setdiff(overlap_genes, ref_genes))
```

```
## [1] 59
```

```r
# number of targets unique to database
length(base::setdiff(ref_genes, overlap_genes))
```

```
## [1] 10803
```