BS_TSS_Distance_OmpR_v1.0

```
## SET WORKING DIRECTORY
setwd("/Users/laura/Documents/PGC/BS-TSS-distances/")
```

OmpR DATA

REGULON-DB

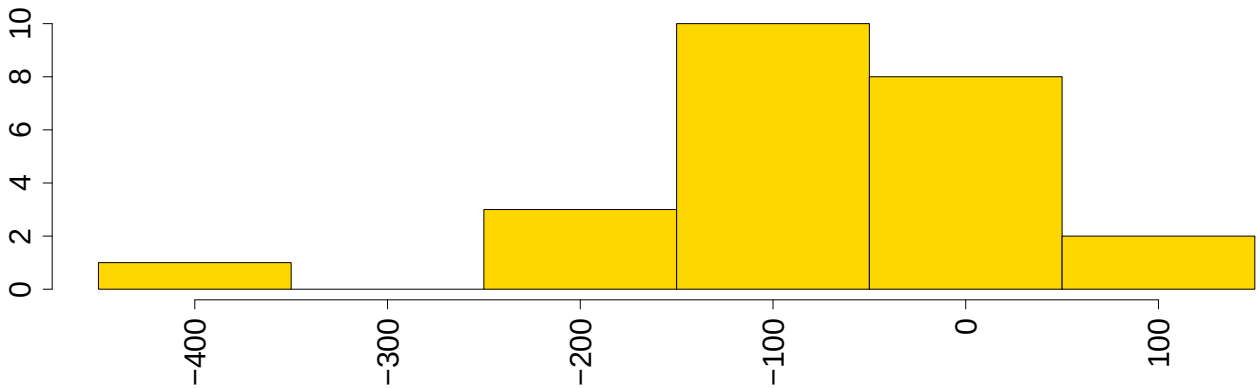
```
bs.rdb <- read.table("Example-Data/INPUT/BindingSiteSet.txt", header=F, sep="\t", stringsAsFactors = F)
names(bs.rdb) <- c("TF.ID", "TF", "TFBS.ID", "TF.LEFT", "TF.RIGTH", "TFBS.SATRND", "TF.GENE.ID", "TU", "

bs.ompr <- subset(bs.rdb, TF == "OmpR")
dim(bs.ompr)
```

```
## [1] 25 14
```

Distance from TFBS to TSS

ALL TFBS

```
{hist(bs.ompr$DIST.TSS, breaks = seq(-450, 150, by = 100), main = "",
     xlab = "", ylab = "", col = "gold", cex.axis=2, cex.lab = 3, xaxt="n")
axis(1, at=seq(-400, 100, by = 100), labels=seq(-400, 100, by = 100), cex.axis=2, las = 2)}
```

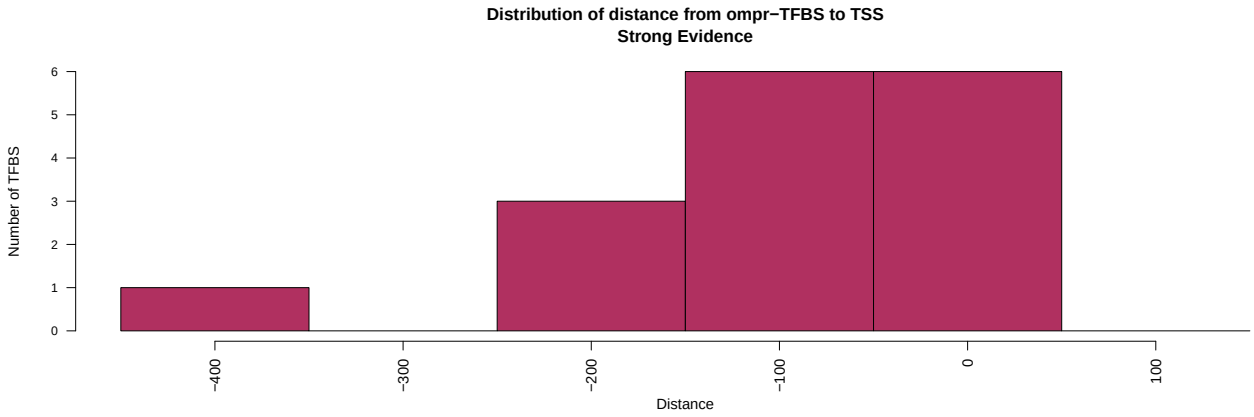


ONLY STRONG TFBS

```
bs.ompr.strong <- subset(bs.ompr, EVIDENCE.LEVEL == "Strong")
dim(bs.ompr.strong)
```

```
## [1] 16 14
```

```
{hist(bs.ompr.strong$DIST.TSS, breaks = seq(-450, 150, by = 100), main = "Distribution of distance from
     xlab = "Distance", ylab = "Number of TFBS", col = "Maroon", cex.axis = 0.8, las = 2, xaxt="n")
axis(1, at=seq(-400, 100, by = 100), labels=seq(-400, 100, by = 100), cex=0.8, las = 2)}
```

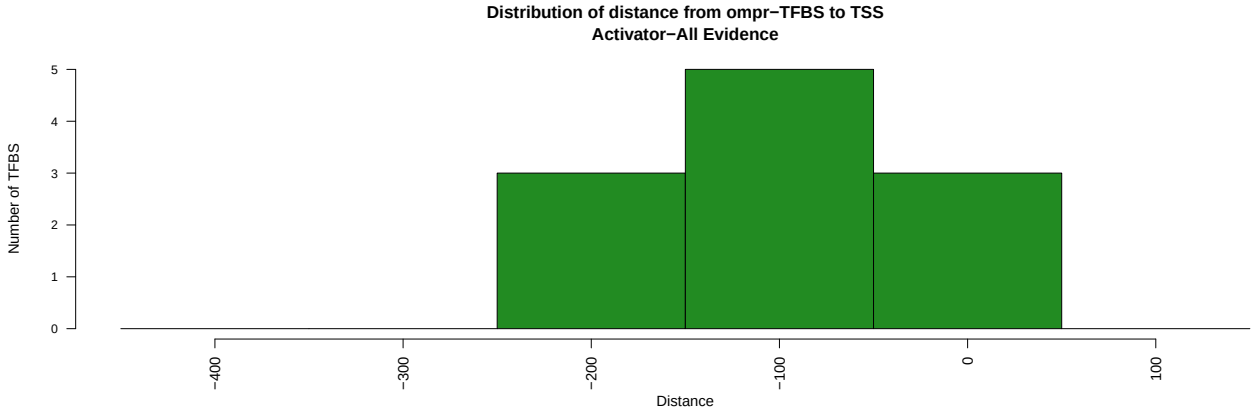


ONLY ACTIVATOR TFBS, ALL EVIDENCE

```
bs.ompr.activator <- subset(bs.ompr, EFFECT == "+")
dim(bs.ompr.activator)
```

```
## [1] 11 14
```

```
{hist(bs.ompr.activator$DIST.TSS, breaks = seq(-450, 150, by = 100), main = "Distribution of distance fr
     xlab = "Distance", ylab = "Number of TFBS", col = "ForestGreen", cex.axis = 0.8, las = 2, xaxt="n"
axis(1, at=seq(-400, 100, by = 100), labels=seq(-400, 100, by = 100), cex=0.8, las = 2)}
```

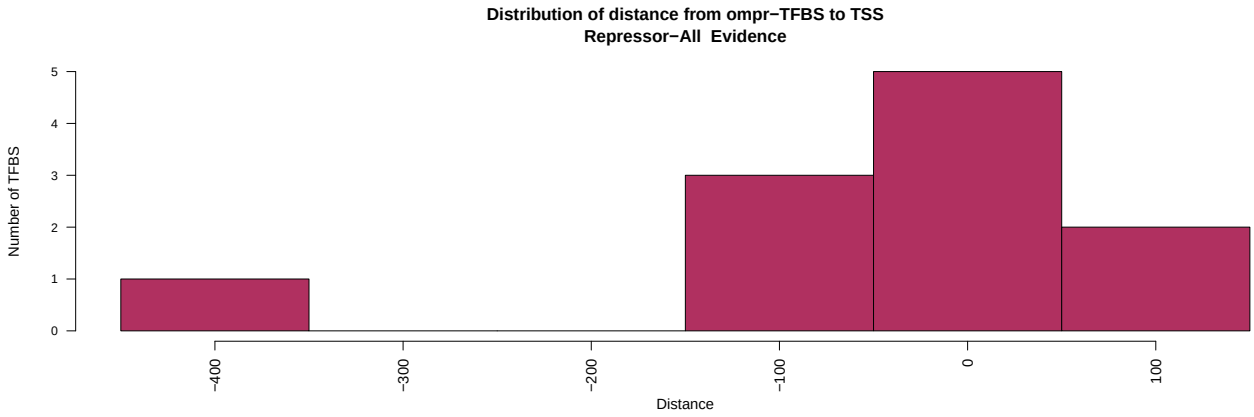


ONLY REPRESSOR TFBS, ALL EVIDENCE

```
bs.ompr.repressor <- subset(bs.ompr, EFFECT == "-")
dim(bs.ompr.repressor)
```

```
## [1] 12 14
```

```
{hist(bs.ompr.repressor$DIST.TSS, breaks = seq(-450, 150, by = 100), main = "Distribution of distance fr
     xlab = "Distance", ylab = "Number of TFBS", col = "Maroon", cex.axis = 0.8, las = 2, xaxt="n")
axis(1, at=seq(-400, 100, by = 100), labels=seq(-400, 100, by = 100), cex=0.8, las = 2)}
```



HOW MANY BS ARE AT DISTANCE X OR LOWER

ACTIVATORS

```
quantile(bs.ompr.activator$DIST.TSS, probs = seq(0, 1, 0.1))
```

```
##     0%    10%    20%    30%    40%    50%    60%    70%    80%    90%
## -206.5 -186.5 -165.5  -90.5  -88.5  -71.5  -70.5  -67.5  -49.5  -47.5
##   100%
##  -45.5
```

80% of the activator BS's are at a distance between -186.5 and -47.5

REPRESSORS

```
quantile(bs.ompr.repressor$DIST.TSS, probs = seq(0, 1, 0.1), na.rm = TRUE)
```

```
##     0%    10%    20%    30%    40%    50%    60%    70%    80%    90%
## -370.5 -149.5 -145.5  -60.5  -47.5  -45.5  -31.5   18.5   18.5   69.5
##   100%
##  119.5
```

80% of the repressor BS's are at a distance between -149.5 and +69.5

GALAGAN CHIP-SEQ EXPERIMENTS. ompr DATA

PEAKS IN REGULON-DB

FILE: ../Data/OmpR-Galagan/OmpR-Known.csv DOWNLADED FROM: OmpR EXPERIMENTS, Known Transcription Factor Binding Sites for ompr (By Position)

```
known.galagan <- read.csv("Example-Data/INPUT/OmpR-Known.csv", header=F, stringsAsFactors = F)
names(known.galagan) <- c("Peak.Found", "ID1", "TF", "ID2", "Left.End", "Right.End", "Center", "Strand"
dim(known.galagan)
```

```
## [1] 25 17
```

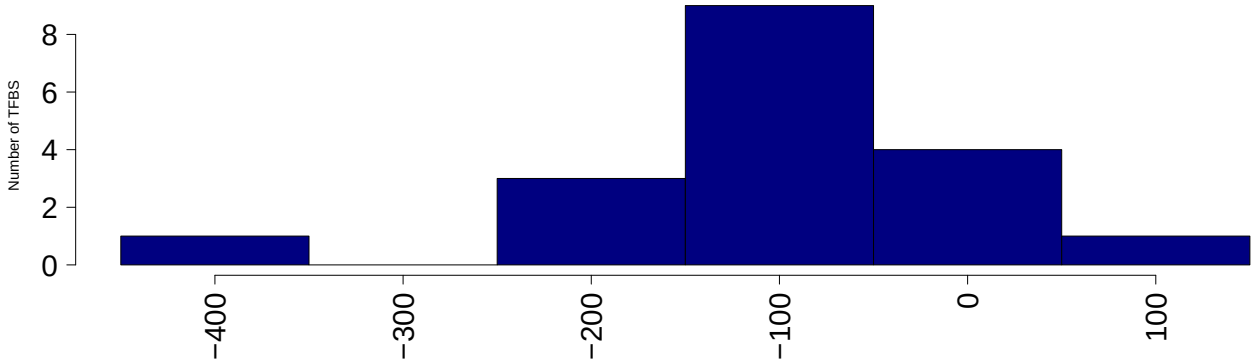
Plot only those with a known distance to TSS

```
known.galagan <- subset(known.galagan, !is.na(Promoter.Position))
dim(known.galagan)
```

```
## [1] 24 17
```

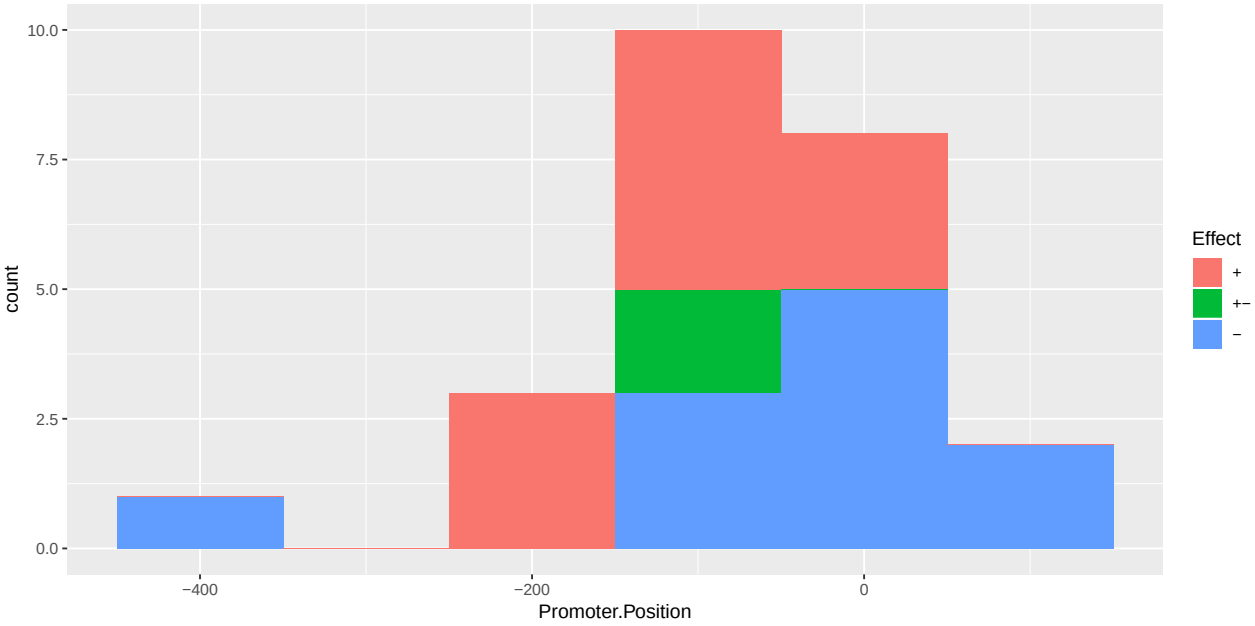
Distribution of TFBS to TSS

```
{ hist(known.galagan$Promoter.Position[known.galagan$Peak.Found > 0], breaks = seq(-450, 150, by = 100)
     #main = "Distribution of distance from TFBS to TSS\nGalagan Known TFBS OmpR",
     main = "",
     xlab = "", ylab = "Number of TFBS", col = "Navy", cex.axis = 2, las = 2, xaxt="n")
axis(1, at=seq(-400, 100, by = 100), labels=seq(-400, 100, by = 100), cex.axis=2, las =2) }
```



Classify by effect each BS (ALL TFBS)

```
ggplot(known.galagan,aes(x=Promoter.Position,fill=Effect))+geom_histogram(position="stack", binwidth =
```



APPLICATIONS

- Identify most probable BSs based on the distance TSS-BS for genes with known TSS
- Identify most probable sets BS-TSSs for genes with unknown TSS

Identify most probable BSs based on the distance TSS-BS for genes with known TSS

Read information for all CHIP-SEQ Peaks with RNA-SEQ Data. OmpR DATA

These information associates an Effect with a Target Gene.

TABLE: RNA-Seq Data for Peak Targets

```
peak <- read.csv("Example-Data/INPUT/OmpR-RNASeq-PeakTargets.csv", header=F, stringsAsFactors = F)
names(peak) <- c("Run", "Sample", "TF", "Target", "LogFoldFPKM",  "LogFoldTPM", "FPKM", "Counts","WildTy

peak$effect[peak$LogFoldFPKM > 0] <- "+"
peak$effect[peak$LogFoldFPKM < 0] <- "-"
```

Read information for all ChIP-Seq Peaks

These information associates a Target Gene with a set of BSs.

FILE: ../Data/OmpR-Galagan/OmpR-All.csv DOWNLADED FROM: OmpR EXPERIMENTS,ChipSeq Transcription Factor Binding Sites and Interactions for OmpR

```
all <- read.csv("Example-Data/INPUT/OmpR-All.csv", header=F, stringsAsFactors = F)
names(all) <- c("Exp", "sample", "TF", "type", "ID1", "Start", "Stop", "PeakPos", "Height", "No1", "No2
```

Read TSS information (RegulonDB)

These information associates a TSS with a gene.

FILE TSS: ../Data/pm_w_first_transc_g___w_ids_noHT.txt
Provided by Hely: pm_w_first_transc_g___w_ids.txt **grep -v "TSS_" pm_w_first_transc_g___w_ids.txt > pm_w_first_transc_g___w_ids_noHT.txt**

```
tss <- read.table("Example-Data/INPUT/pm_w_first_transc_g__w_ids_noHT.txt", header=F, stringsAsFactors =
names(tss) <- c("ID", "Promoter.Name", "TSS", "Sigma", "Strand", "GI", "Gene", "PosLeft", "PosRigth", "I
tss <- subset(tss, !is.na(TSS) )
```

Calculate distance TSS-BS

For every gene (separate activated or repressed genes):

Calculate number of TSSs per gene

```
gene.repressed <- peak$Target[which(peak$effect == "-")]
gene.activated <- peak$Target[which(peak$effect == "+")]

length(gene.repressed)
```

```
## [1] 128
```

```
length(gene.activated)
```

```
## [1] 30
```

```
no.tss.repressed <- sapply(gene.repressed, function(x,tss){ nrow(subset(tss, Gene == x))}, tss = tss, si
no.tss.activated <- sapply(gene.activated, function(x,tss){ nrow(subset(tss, Gene == x))}, tss = tss, si
```

For all genes with at least one associated TSS:
* Look all BS ChIP-Seq associated with that gene * Calculate distance between each BS and each TSS

```
calculate_distance <- function(x, tss, bs){
  bs.gene <- subset(bs, Gene == x)
  tss.gene <- subset(tss, Gene == x)
  if (nrow(tss.gene) > 0){
    distance.all <- sapply(bs.gene$PeakPos, function(x,tss){
```

```
    if (tss.gene$Strand[1] == "forward"){
      distance <- tss.gene$TSS - x
    }else{
      distance <- x - tss.gene$TSS
    }
    distance
    } ,simplify = T, tss = tss)
  }else{
    return (NA)
  }
}

TSS.activated <- sapply(gene.activated[no.tss.activated > 0], calculate_distance, tss = tss, bs = all, s
TSS.repressed <- sapply(gene.repressed[no.tss.repressed > 0], calculate_distance, tss = tss, bs = all, s

dist.TSS.activated <- unlist(TSS.activated)
dist.TSS.repressed <- unlist(TSS.repressed)

length(dist.TSS.activated)
```

```
## [1] 31
```

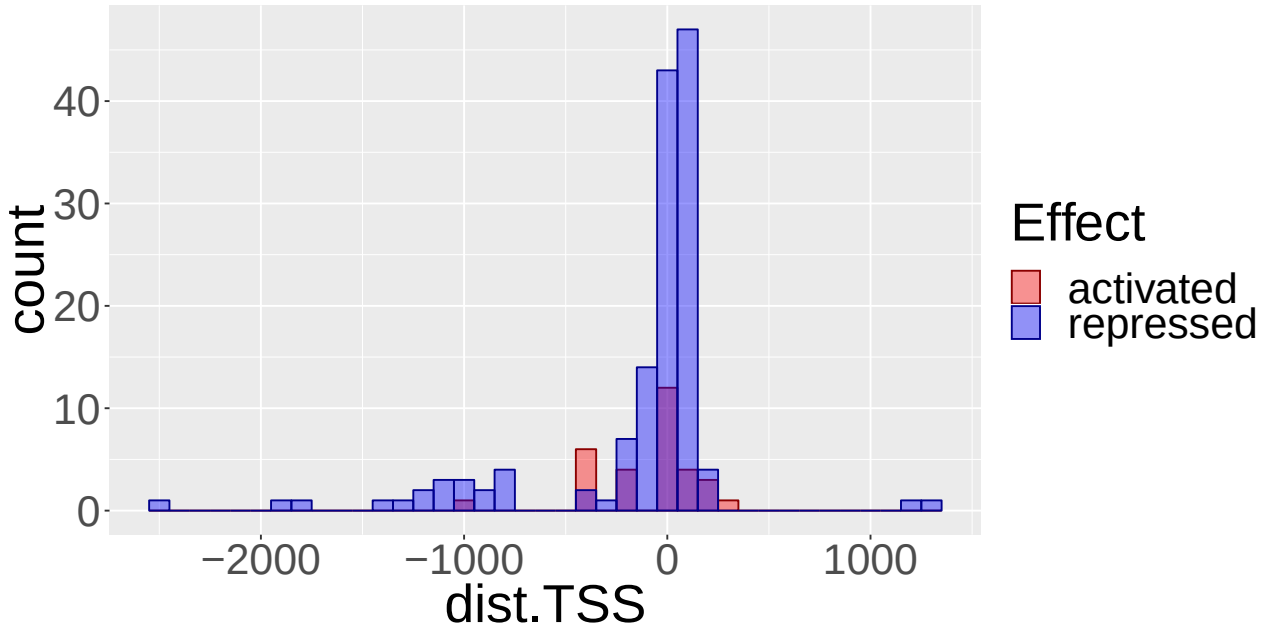
```
length(dist.TSS.repressed)
```

```
## [1] 139
```

Number of Distribution of Distance from TFBS to TSS classified by Effect

```
dist.TSS <- data.frame(gene = c(names(dist.TSS.activated), names(dist.TSS.repressed)), dist.TSS = c(dis

ggplot(dist.TSS,aes(x=dist.TSS,fill=Effect)) +geom_histogram(position="stack", binwidth = 100) + theme(
```

```
ggplot(dist.TSS,aes(x=dist.TSS, fill = Effect, col = Effect)) + geom_histogram(alpha = 0.4, position =
```



Example iraP

TWO TSSs

```
subset(tss, Gene == "iraP")
```

```
##                ID Promoter.Name    TSS   Sigma  Strand GI Gene PosLeft
## 1153 ECK120030215        iraPp1 401319 Sigma70 forward NA iraP  401386
## 1154 ECK120034408        iraPp2 401363         forward NA iraP  401386
##      PosRigth
## 1153   401646
## 1154   401646
##
## 1153 Non-traceable author statement,Transcription initiation mapping,RNA-seq using two enrichment st
## 1154                High-throughput transcription initiation mapping,RNA-seq using two enrichment st
##      Bnumber
## 1153   b0382
## 1154   b0382
##                                                                                          PromSeq
## 1153 gctggtaatcaaacaaaaaatatttgcgcaaagtatttcctttgtcataaaaataatactTccagacactatgaagttgtg
## 1154 tcataaaaataatacttccagacactatgaagttgtgaaacataatgttaacttctccatActttggataaggaaatacag
##      DistPromGene
## 1153          -67
## 1154          -23
```

TWO BINDING SITES

```
subset(all, Gene == "iraP")
```

```
##                    Exp    sample   TF      type    ID1  Start   Stop
## 43 Acid_Ecoli_Rerun_V3 OmpR_91_4 ompR inducible U00096 401263 401445
## 44 Acid_Ecoli_Rerun_V3 OmpR_91_4 ompR inducible U00096 401629 401831
##    PeakPos Height  No1      No2 Shift Bnumber Gene Dist NA
```

```
## 43  401344    846 2.36 0.018165    57   b0382 iraP    0 NA
## 44  401752    859 2.40 0.015588    62   b0382 iraP    0 NA
```

DISTANCE BETWEEN EACH BS AND EACH TSS

- Una columna para cada BS
- Una fila para cada TSS

```
TSS.activated[['iraP']]
```

```
##      [,1] [,2]
## [1,]  -25 -433
## [2,]   19 -389
```

80% of the activator BS's are at a distance between -186.5 and -47.5

CONCLUSION:
THE FIRST BINDING SITE HAS A BETTER CHANCE TO BE FUNCTIONAL THAN THE SECOND ONE
THE FIRST BINDING SITE IS PROBABLY RELATED WITH THE FIRST TSS

Filter BS-TSS interactions

Repressed genes

Number of ALL TSS-BS interactions

```
sapply(TSS.repressed, function(x){length(as.vector(x))}, simplify=T)
```

```
## aaeR betI csgD cspA cspE dhaR exuR flhD galR gcvA pdhR prpR  can fadE yagK
##    1    1    6    2    2    1    1    4    2    1    1    1    1    1    2
## betT yaiZ phoA  tsx cyoA cstA citC lipA chiP gltA glnH ompX ompF asnS yccT
##    1    1    1    2    1    6    2    2    2    2    2    4    3    2    2
## dhaK narU hdhA dtpA slyB infC znuA znuC gatZ gatY ompC nuoA sixA fadL galP
##    1    4    1    3    2    2    1    1    1    1    9    2    2    1    2
## metC yghB patA ygjR sstT yqjA rimP arcB ppiA ugpB livK rpoH yhiM hdeD dppA
##    1    1    1    2    2    2    1    1    4    2    2    5    1    2    1
## atpI ysgA nrfA dcuA miaA  hfq purA ytfK fecI creA csiD
##    1    2    1    2    2    2    1    1    2    2    2
```

```
sum(sapply(TSS.repressed, function(x){length(as.vector(x))}, simplify=T))
```

```
## [1] 139
```

```
length(TSS.repressed)
```

```
## [1] 71
```

Number of possible TSS-BS interactions (AFTER FILTERING)

```
pass.repressed <- lapply(TSS.repressed, function(x, min, max){ x[x>min & x<max]}, min = -250 , max = 50

#pass.repressed <- lapply(TSS.repressed, function(x, min, max){ x[x>min & x<max]}, min = -150 , max = 7

no.repressed <- sapply(pass.repressed, length, simplify=T)
no.repressed
```

```
## aaeR betI csgD cspA cspE dhaR exuR flhD galR gcvA pdhR prpR  can fadE yagK
##    0    0    0    2    2    1    0    4    0    0    1    0    0    1    0
## betT yaiZ phoA  tsx cyoA cstA citC lipA chiP gltA glnH ompX ompF asnS yccT
```

```
##    1    1    1    2    1    6    2    0    1    1    1    2    0    0    2
## dhaK narU hdhA dtpA slyB infC znuA znuC gatZ gatY ompC nuoA sixA fadL galP
##    1    0    0    3    2    0    1    0    1    0    2    1    0    1    0
## metC yghB patA ygjR sstT yqjA rimP arcB ppiA ugpB livK rpoH yhiM hdeD dppA
##    0    1    0    0    1    1    1    0    4    1    2    1    0    0    0
## atpI ysgA nrfA dcuA miaA  hfq purA ytfK fecI creA csiD
##    0    2    0    0    0    1    1    0    0    1    2
```

```
sum(no.repressed)
```

```
## [1] 63
```

Activated genes

Number of ALL TSS-BS interactions

```
sapply(TSS.activated, function(x){length(as.vector(x))}, simplify=T)
```

```
## malT iraP ybgI cydA mcbA ycgZ ychH ydhI rpmI ftnB glpA raiA rlmD hdeA xylF
##    2    4    2    5    2    2    2    1    1    2    1    1    1    1    1
## malK aspA
##    2    1
```

```
sum(sapply(TSS.activated, function(x){length(as.vector(x))}, simplify=T))
```

```
## [1] 31
```

```
length(TSS.activated)
```

```
## [1] 17
```

Number of possible TSS-BS interactions (AFTER FILTERING)

```
pass.activated <- lapply(TSS.activated, function(x, min, max){ x[x>min & x<max]}, min = -250 , max = -5

#pass.activated <- lapply(TSS.activated, function(x, min, max){ x[x>min & x<max]}, min = -190 , max = -

no.activated <- sapply(pass.activated, length, simplify=T)
no.activated
```

```
## malT iraP ybgI cydA mcbA ycgZ ychH ydhI rpmI ftnB glpA raiA rlmD hdeA xylF
##    0    0    1    0    0    0    1    0    0    0    0    1    0    0    0
## malK aspA
##    0    1
```

```
sum(no.activated)
```

```
## [1] 4
```

Identify most probable sets BS-TSSs for genes with unknown TSS

Identify genes with no associated TSSs

Genes with no associated TSS

```
activated.noTSS <- names(no.tss.activated[no.tss.activated == 0])
repressed.noTSS <- names(no.tss.repressed[no.tss.repressed == 0])

length(activated.noTSS)
```

```
## [1] 13
```

```
activated.noTSS
```

```
##  [1] "yjhI" "caiD" "yaiS" "ybcV" "yeaI" "yeaN" "gatA" "yehD" "yffL" "ygdI"
## [11] "rpsS" "hdeB" "tnaA"
```

```
length(repressed.noTSS)
```

```
## [1] 57
```

```
repressed.noTSS
```

```
##  [1] "bglJ" "fruR" "yncC" "slyA" "yeaM" "yqeH" "yaaJ" "caiC" "yadG" "proA"
## [11] "yagU" "psiF" "yaiC" "rhsD" "tatE" "elfD" "ycfH" "dauA" "chaA" "yncD"
## [21] "yddB" "yeaH" "prc"  "holE" "yobB" "wzxB" "rfbC" "gatB" "yfaL" "yfaZ"
## [31] "yffQ" "yffS" "ygbJ" "ygbE" "cysC" "ygcE" "queE" "yqcC" "yqiJ" "elbB"
## [41] "gltF" "yhcN" "rplV" "ftsX" "yhiD" "yhjE" "yibB" "waaO" "waaB" "typA"
## [51] "mgtA" "yjgL" "insG" "yjhX" "yjiC" "mdtM" "yccU"
```

Identify TF Binding Sites associated to those genes

```
activated.noTSS.bs <- subset(all, Gene %in% activated.noTSS)
repressed.noTSS.bs <- subset(all, Gene %in% repressed.noTSS)

dim(activated.noTSS.bs)
```

```
## [1] 15 16
```

```
dim(repressed.noTSS.bs)
```

```
## [1] 70 16
```

Look for TSSs (only sense TSSs) in Storz data associated to each of the genes with no associated TSS

- Read Gisella data
- Subset only sense TSSs

FROM GISELLA STORZ DATA: TSS POSITION (*only sense TSS*) FROM JAMES GALAGAN DATA: BINDING SITES POSITIONS, EFFECT OF THE BINDING SITES

FILE:StorzG_TSS_Table_M63_0.4.txt

```
M63 <- read.table("Example-Data/INPUT/StorzG_TSS_Table_M63_0.4.txt", stringsAsFactors = F, header=F, se
names(M63) <- c("TSSPosition", "RPKM", "Promoter", "Strand", "RelPos", "Gene", "Bnumber", "LeftGene", "
type.sense <- c("intragenic/sense", "upstream/sense")
sense <- subset(M63, Orientation %in% type.sense)
```

- Count the number of possible associated TSSs in Storz data

Activated genes

```
tss.no.activated <- sapply(activated.noTSS, function(x,sense){ nrow(subset(sense, Gene == x))}, sense =
#tss.no.activator
```

Repressed genes

```
tss.no.repressed <- sapply(repressed.noTSS, function(x,sense){ nrow(subset(sense, Gene == x))}, sense =
#tss.no.repressor
```

For all genes with at least one associated TSS (from Storz data):
* Look all BS ChIP-Seq associated with that gene * Calculate distance between each BS and each TSS

```
TSS_BS_distance<- function(x, tss, bs){
  bs.gene <- subset(bs, Gene == x)
  tss.gene <- subset(tss, Gene == x)
  if (nrow(tss.gene) > 0){
    distance.all <- sapply(bs.gene$PeakPos, function(x,tss.gene){
      if (tss.gene$Strand[1] == "-"){
        distance <- tss.gene$TSSPosition - x
      }else{
        distance <- x - tss.gene$TSSPosition
      }
      distance
    }, tss.gene = tss.gene, simplify=T)
    return(distance.all)
  }else{
    return (NA)
  }
}
```

```
TSS_BS.dist.activated <- sapply(activated.noTSS[tss.no.activated > 0], TSS_BS_distance, tss=sense, bs=al
TSS_BS.dist.repressed <- sapply(repressed.noTSS[tss.no.repressed > 0], TSS_BS_distance, tss=sense, bs=al
```

Example: ydbR

THREE TSSs

```
subset(sense, Gene == "yaaJ")
```

```
##    TSSPosition   RPKM                                              Promoter
## 16        7482  48.04 gcgttctgttcgccgtctttttgctcatcgcctatggcataattttcagcG
## 19        8053 301.05 cgttgatatcgcccgccatttttatacaaaacctcatgtatgctacgcagA
## 20        8226 298.58 ggtggtgtactgacgaagggaggtcaatttgtccgtcatgatagtatttcT
##    Strand RelPos Gene Bnumber LeftGene RigthGene      Orientation
## 16      -    477 yaaJ   b0007     6529      7959 intragenic/sense
## 19      -    -94 yaaJ   b0007     6529      7959   upstream/sense
## 20      -   -267 yaaJ   b0007     6529      7959   upstream/sense
##     TSSClass Enrichment       evidence
## 16  internal          1 ICA,RS-EPT-CBR
## 19   primary          1 ICA,RS-EPT-CBR
## 20 secondary          1 ICA,RS-EPT-CBR
```

ONE BINDING SITE

```
subset(all, Gene == "yaaJ")
```

```
##                    Exp    sample   TF      type    ID1 Start Stop PeakPos
## 26 Acid_Ecoli_Rerun_V3 OmpR_91_4 ompR inducible U00096  7927 8097    8017
##    Height  No1     No2 Shift Bnumber Gene Dist NA
## 26    937 2.62 0.01249    67   b0007 yaaJ    0 NA
```

DISTANCE BETWEEN EACH BS AND EACH TSS

- Una columna para cada BS
- Una fila para cada TSS

```
TSS_BS.dist.repressed[['yaaJ']]
```

```
##      [,1]
## [1,] -535
## [2,]   36
## [3,]  209
```

80% of the repressor BS's are at a distance between -149.5 and +69.5

CONCLUSION:
THE SECOND TSS HAS A BETTER CHANCE TO BE FUNCTIONAL THAN ALL THE OTHERS
THE SECOND TSS COULD BE RELATED TO THE ONLY BINDING SITE REPORTED

Filter BS-TSS interactions

Repressed genes

Number of ALL TSS-BS interactions

```
sapply(TSS_BS.dist.repressed, function(x){length(as.vector(x))}, simplify=T)
```

```
## bglJ slyA yeaM yqeH yaaJ yadG proA yagU rhsD tatE ycfH dauA chaA yncD yeaH
##    2    2    2    1    3    1    1    2    5    2    4    6    6    5    1
##  prc holE yobB wzxB gatB yfaL yfaZ yffS ygbJ ygbE cysC queE yqcC yqiJ elbB
##    1    2    6    2    1    5    2    2    1    2    2    2    1    1    1
## gltF yhcN ftsX yhjE yibB waaB typA mgtA yjgL insG yjhX mdtM yccU
##    4    1    1    4    4    1    8    2    4    5    2    2    2
```

```
sum(sapply(TSS_BS.dist.repressed, function(x){length(as.vector(x))}, simplify=T))
```

```
## [1] 114
```

Number of possible TSS-BS interactions (AFTER FILTERING)

```
pass.repressor <- lapply(TSS_BS.dist.repressed, function(x, min, max){ x[x>min & x<max]}, min = -250 , m


#pass.repressor <- lapply(TSS_BS.dist.repressed, function(x, min, max){ x[x>min & x<max]}, min = -200 ,
no.repressor <- sapply(pass.repressor, length, simplify=T)
no.repressor
```

```
## bglJ slyA yeaM yqeH yaaJ yadG proA yagU rhsD tatE ycfH dauA chaA yncD yeaH
##    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0
##  prc holE yobB wzxB gatB yfaL yfaZ yffS ygbJ ygbE cysC queE yqcC yqiJ elbB
##    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## gltF yhcN ftsX yhjE yibB waaB typA mgtA yjgL insG yjhX mdtM yccU
##    0    0    0    0    0    0    0    1    0    0    0    0    0
```

```
sum(no.repressor)
```

```
## [1] 2
```

Activated genes

Number of ALL TSS-BS interactions

```
sapply(TSS_BS.dist.activated, function(x){length(as.vector(x))}, simplify=T)
```

```
## yjhI yaiS yeaI yeaN yffL ygdI tnaA
##    1    1    2    1    1    2    4
```

```r
sum(sapply(TSS_BS.dist.activated, function(x){length(as.vector(x))}, simplify=T))
```

```
## [1] 12
```

Number of possible TSS-BS interactions (AFTER FILTERING)

```r
pass.activated <- lapply(TSS_BS.dist.activated, function(x, min, max){ x[x>min & x<max]}, min = -250 , m

#pass.activated <- lapply(TSS_BS.dist.activated, function(x, min, max){ x[x>min & x<max]}, min = -460 ,
no.activated <- sapply(pass.activated, length, simplify=T)
no.activated
```

```
## yjhI yaiS yeaI yeaN yffL ygdI tnaA
##    0    0    0    0    0    0    0
```