April Markdown

Awais Choudhry

03/04/2017

### 2-4-2017

Today i took some data from my previous table values and combined it with Tessa's prervious data, specifically interested in finding if there is a correlation between distance to cloest Arx TFBS and the amount of differential expression. Intially i found no correlation as expected as DNA is not actually linear within the cell.

However, i would be interesting to see if there is distance may help deregulate Arx. Note the first chunk of code is needed to generate the table i need to excute my new code hence i placed it above my code so the chunk of code here is able to be excuted from start to finish alone.

# Old Code needed to excute my new code below  
  
 library(readr)  
 library(dplyr)  
library(Biostrings)  
library(GenomicRanges)  
library(BSgenome.Mmusculus.UCSC.mm9)  
  
#importing the relevant Data  
   
 geneSymbols <- read\_tsv("~/DataFiles/Tessa differential Expressed/2colmultiplenames.txt")  
 pointer<-import(con= "~/DataFiles/Tessa differential Expressed/PA16col.bed")%>%as.data.frame()  
 enhancerDataFrame<- import( con = "~/DataFiles/Enhancer Tracks/Mouse/Enhanceresmm9.bed")%>%ranges()%>%as.data.frame()  
 differentiallyExpressed <- left\_join(pointer, geneSymbols, by = c("name"="#kgID"))  
 geneNames<- as.data.frame(unique(geneSymbols[2]))#subsetting for intergert  
 arx6Mer <-  
 rbind(  
 A = c(0, 1, 1, 0, 0, 1),  
 C = c(0, 0, 0, 0, 0, 0),  
 G = c(0, 0, 0, 0, 0, 0) ,  
 T = c(1, 0, 0, 1, 1, 0)  
 )  
   
 mouseArxDataFrame<-matchPWM(arx6Mer, BSgenome.Mmusculus.UCSC.mm9,"100%")%>%as.data.frame  
   
   
 x<-1#intializing X  
 ArxTable3<- NULL  
 dataFrameConversation<-dataFrameGrange  
 for(i in 1:dim(geneNames[1])){  
   
   
 selectedGene<-differentiallyExpressed[differentiallyExpressed$geneSymbol==as.character(geneNames[x,1]),]#this will select for each gene ID from the differentially expressed data  
 toM<-max(selectedGene[3])+10000##max to get the longest transcript distance  
 fromM<-min(selectedGene[2])-10000##to get the max utilising this as a promoter   
 chrM<-selectedGene[1,1]##plotting the chromosome! this will be fucked need to revalute the chromosme  
   
   
 ##closet ARX motif, done via subtraction of gene start site - ARX data frame and locating the closest TFBS  
   
 ##distance to ARX motif, Select for motif starts and the start site of gene find the lowest value  
 tableDistanceToArxTFBS<- min(abs(mouseArxDataFrame[2]-min(selectedGene[2])))## utlising min and abs to get the closest DataFrame distnace  
   
 #remove the NA from the Cluster Data frame ##i shoudl rename this ;')   
 ##find distance between the start site of gene and the lo  
 grangeDataFrame3<-grangeDataFrame2[-1,]  
 tableDistanceToArxCluster<-min(abs(grangeDataFrame3[1]-min(selectedGene[2])),  
 na.rm = FALSE)  
 ##distance of motifs to enhancer?  
 tableDistanceToEnhancer<- min(abs(enhancerDataFrame-(min(selectedGene[2])+tableDistanceToArxTFBS)))  
   
 ## okay so this is gonna be confusing but simply   
 tableConservationScoreOfTFBS<-dataFrameConversation$score[abs(min(selectedGene[2]+tableDistanceToArxTFBS)-dataFrameConversation[2])==min(abs(min(selectedGene[2]+tableDistanceToArxTFBS)-dataFrameConversation[2]))]  
 tableConservationScoreOfCluster<-dataFrameConversation$score[abs(min(selectedGene[2]+tableDistanceToArxCluster)-dataFrameConversation[2])==min(abs(min(selectedGene[2]+tableDistanceToArxCluster)-dataFrameConversation[2]))]  
   
 if(!exists("ArxTable3"))##if the ArxTable3 doesn't exist it will create it   
 {  
   
 ArxTable3<- cbind("Gene"=selectedGene[1,8],  
 "Chromosome" = as.character(selectedGene[1,1]),  
 "Start"= min(selectedGene[2]),  
 "End"=max(selectedGene[3]),   
 "Distance to closest Arx TFBS"=tableDistanceToArxTFBS,  
 "Conservation score of TFBS"= tableConservationScoreOfTFBS,  
 "Distance to closest cluster"=tableDistanceToArxCluster,  
 "Conservation score of ARX cluster"=tableConservationScoreOfCluster)  
 } else {## if it does exist it will excute and add theac line usinig R line  
 ArxTable3<- rbind(ArxTable3,cbind(  
 "Gene"=selectedGene[1,8],  
 "Chromosome" = as.character(selectedGene[1,1]),  
 "Start"= min(selectedGene[2]),  
 "End"=max(selectedGene[3]),   
 "Distance to closest Arx TFBS"=tableDistanceToArxTFBS,  
 "Conservation score of TFBS"= tableConservationScoreOfTFBS,  
 "Distance to closest cluster"=tableDistanceToArxCluster,  
 "Conservation score of ARX cluster"=tableConservationScoreOfCluster))  
   
 }  
 x<- x+1  
 print(x)   
 }  
   
   
   
##NEW CODE STARTS HERE  
   
### Combinding into new Table  
library(readxl)  
tessaData <- read\_excel("~/Research Proposal/Supplementary\_tables\_1-3.xlsx",   
 skip = 1)%>% as.data.frame()  
  
ArxTable3DataFrame<- as.data.frame(ArxTable3)  
rnaSeqData <- left\_join(ArxTable3DataFrame, tessaData, by = c("Gene"="Gene ID"))  
newTable<-cbind(rnaSeqData, "absoulte of log FC"= abs(rnaSeqData$logFC) )  
  
##plotting the new Table and seeing that there is not correlation!  
library(ggplot2)  
ggplot(data= newTable, aes(x=`Distance to closest Arx TFBS`, y=`logFC` ) )+  
 geom\_jitter()+  
 geom\_point()+  
 theme\_bw()  
  
ggplot(data= newTable, aes(x=`Distance to closest Arx TFBS`, y=`logFC` ) )+  
 geom\_jitter()+  
 geom\_point()+  
 theme\_bw()  
#no relationship between the distance and the differential expression!

To do this i began by importing Tessa's data and converting it to a dataFrame, this will allow me to join it to my own dataframe containing arx motifs and their distances to genes they regulate and allow ggplot to read it. Second thing i did was take the ArxTable3 and turn it into a dataframe which was then joined with Tessa's data based on gene symbol names. This showed me that many of the UCSC ID names i have taken were miss convereted hence resulting in no matches. This will need to be rectified, either i will ultiise a new gene converted or i will change everytyhing away from the UCSC gene IDs. This new table is called "rnaSeqData" and it retunred NA's to coloumns where there was no data. I plotted distance from TSS to the start of the Arx motif to see if there is a differenc to examine the correlation between Arx differential expression and

## 6-4-2017

After my consulatation with Cheryl on the 4-4-2017 i am attempting to generate some scripts that will identify the proportion of Arx motifs in; a gene, an exon, an ehnacner, methylation sites etc. The intention here is to see if a particular motif is more enriched in a promoters and enhancers over non-coding regions to see if it is more functionally significant. The three motif models i'll be testing is the arx 6 mer model developed by Berger et al (2008) and confirmed by selex and in vivo experiment since. In addition, i will be testing jolma's CAP-SELEX mechanism, and the core of the 6mer, which is the TAAT.

##Productive R DAY  
  
  
library(Biostrings)  
library(GenomicRanges)  
library(pander)  
library(rtracklayer)  
library(GenomicFeatures)  
library(MotifDb)  
library(BSgenome.Mmusculus.UCSC.mm9)  
library(magrittr)  
library(GenomicFeatures)  
library(ggplot2)  
library(reshape2)  
  
gtfUCSCexonscoding<-import("~/Scripts/March/FullMm9genome.GTF")  
gtfUCSCgenes<- import("~/Scripts/March/mm9.bed")  
enhancerGrange <- import( con = "~/DataFiles/Enhancer Tracks/Mouse/Enhanceresmm9.bed")  
methylationGrange<- import(con = "~/DataFiles/Methylation Tracks/Mouse/CpGIslands.bed")  
promoterGrange<- promoters(gtfUCSCgenes)  
genomeToAlignTo<-BSgenome.Mmusculus.UCSC.mm9  
## Getting the other motifs  
arx6Mer<-  
 rbind(  
 A = c(0, 1, 1, 0, 0, 1),  
 C = c(0, 0, 0, 0, 0, 0),  
 G = c(0, 0, 0, 0, 0, 0) ,  
 T = c(1, 0, 0, 1, 1, 0)  
 )  
  
arx6MerTFBS<- matchPWM(arx6Mer, genomeToAlignTo, "100%")  
  
arx4Mer<-rbind(A=c(1, 1, 0,0),  
 C=c(0,0,0,0),   
 G=c(0,0,0,0),  
 T=c(0,0,1,1))  
arx4MerTFBS<-matchPWM(arx4Mer, genomeToAlignTo, "100%")  
  
#arxJolma<-MotifDb::query(MotifDb, "Mmusculus-jolma2013-Arx")[[1]]  
#arxJolma <- round(arxJolma\*100)  
#arxJolmaTFBS<- matchPWM(arxJolma, BSgenome.Mmusculus.UCSC.mm9, "90%", with.score = TRUE)  
  
  
  
  
## make me some pie charts  
arx6MerInsideExons<-length(findOverlaps(gtfUCSCexonscoding,arx6MerTFBS))  
arx6MerInsideGenes<-length(findOverlaps(gtfUCSCgenes, arx6MerTFBS))  
arx6MersInsideEnhancers<- length(findOverlaps(enhancerGrange, arx6MerTFBS))  
arx6MerInsideMethylation<- length(findOverlaps(methylationGrange, arx6MerTFBS))  
arx6merInsidePromoter<- length(findOverlaps(promoterGrange, arx6MerTFBS))  
## to find non-coding TFS.   
exonsArx<-subset(arx6MerTFBS,findOverlaps(gtfUCSCexonscoding,arx6MerTFBS)%>%countRnodeHits())  
genesArx<-subset(arx6MerTFBS,findOverlaps(gtfUCSCgenes,arx6MerTFBS)%>%countRnodeHits())  
doubleCounted6mers<-subset(genesArx, findOverlaps(exonsArx,genesArx)%>%countRnodeHits())  
methylationGrange<-  
  
#arxJolmaInsideExon<- length(findOverlaps(gtfUCSCexonscoding, arxJolmaTFBS))  
#arxJolmaInsideGenes<- length(findOverlaps(gtfUCSCgenes, arxJolmaTFBS))  
#arxJolmaInsideEnhancer<- length(findOverlaps(enhancerGrange, arxJolmaTFBS))  
#arxJolmaInsideMethylation<- length(findOverlaps(methylationGrange, arxJolmaTFBS))  
#arxJolmaInsidePromoter<- length(findOverlaps(promoterGrange, arxJolmaTFBS))  
  
arx4merInsideExon<- length(findOverlaps(gtfUCSCexonscoding, arx4MerTFBS))  
arx4merInsideGenes<- length(findOverlaps(gtfUCSCgenes, arx4MerTFBS))  
arx4merInsideEnhancer<- length(findOverlaps(enhancerGrange, arx4MerTFBS))  
arx4merInsideMethylation<- length(findOverlaps(methylationGrange, arx4MerTFBS))  
arx4merInsidePromoter<- length(findOverlaps(promoterGrange, arx4MerTFBS))  
numberOfOtherGrang4mere<-findOverlaps(arx4MerTFBS, Other)  
##Data table  
table6mer<- cbind("6 mer Motifs in Other Regions" = length(numberOfOtherGrange6mer),  
 "6 mer Motifs Inside Genes"= arx6MerInsideGenes,  
 "6 mer Motifs Inside Exons"=arx6MerInsideExons,  
 "6 mer Motifs Inside Enhancers"=arx6MersInsideEnhancers,  
 "6 mer Motifs Inside CpG Islands"= arx6MerInsideMethylation,  
 "6 mer Motif Inside Promoter Site" =arx6merInsidePromoter)  
  
#tableJolma<- cbind("Jolma Motifs In Total mm9"= length(arxJolmaTFBS),  
# "Jolma Motifs Inside Genes"=arxJolmaInsideGenes ,  
# "Jolma Motifs Inside Exons"= arxJolmaInsideExon,  
# "Jolma Motifs Inside Enhancers" = arxJolmaInsideEnhancer,  
# "Jolma Motifs Inside CpG Islands" = arxJolmaInsideMethylation,  
 # "Jolma Motif Inside Promoter Site" =arxJolmaInsidePromoter)%>%pander()  
  
table4mer<- cbind("4 mer Motifs Inside Other" = length(numberOfOtherGrang4mere),  
 "4 mer Motifs Inside Genes"= arx4merInsideGenes,  
 "4 mer Motifs Inside Exons"=arx4merInsideExon,  
 "4 mer Motifs Inside Enhancers"=arx4merInsideEnhancer,  
 "4 mer Motifs Inside CpG Islands"= arx4merInsideMethylation,  
 "4 mer Motif Inside Promoter Site" =arx4merInsidePromoter)  
 table4merggplot<-t(table4mer)%>%as.data.frame%>%rownames\_to\_column()%>%as.data.frame()  
   
 ## to find non-coding TFS.   
 rownames(table4merggplot)<-c("Other",  
 "Genes",  
 "Exons",  
 "Enhancers",  
 "CpG Islands",  
 "Promoters")  
 table4merggplot<-table4merggplot%>%rownames\_to\_column()%>%as.data.frame()   
 ggplot(table4merggplot, aes(x=rowname, y=V1, fill =rowname))+  
 geom\_bar(stat="identity")+  
 xlab(label= "Genomic Location")+  
 ylab(label= "Number of Arx Motifs")+  
 theme\_bw()+  
 guides(fill=FALSE)+  
 theme(axis.text=element\_text(size=12, angle = 90),  
 axis.title=element\_text(size=14,face="bold"))+  
 geom\_text(size = 5, position = position\_stack(vjust = 0.75), aes(label=V1))  
   
table6merggplot<-t(table6mer)%>%as.data.frame%>%rownames\_to\_column()%>%as.data.frame()  
  
rownames(table6merggplot)<-c("Other",  
 "Genes",  
 "Exons",  
 "Enhancers",  
 "CpG Islands",  
 "Promoters")  
table6merggplot<-table6merggplot%>%rownames\_to\_column()%>%as.data.frame()  
  
ggplot(table6merggplot, aes(x=rowname, y=V1, fill =rowname))+  
 geom\_bar(stat="identity")+  
 xlab(label= "Genomic Location")+  
 ylab(label= "Number of Arx Motifs")+  
 theme\_bw()+  
 guides(fill=FALSE)+  
 theme(axis.text=element\_text(size=12, angle = 90),  
 axis.title=element\_text(size=14,face="bold"))+  
 geom\_text(size = 5, position = position\_stack(vjust = 0.75), aes(label=V1))  
##Cleaning the ChIPseq Data Up!  
n2aChipSeq <- read\_excel("~/DataFiles/ChIPseq/Mouse/ChIPseqDataQuille2011.xls",   
 sheet = "only N2a")%>%as.data.frame  
embyroChipSeq <- read\_excel("~/DataFiles/ChIPseq/Mouse/ChIPseqDataQuille2011.xls", sheet = "only emb brain")%>%as.data.frame  
commonChipSeq <- read\_excel("~/DataFiles/ChIPseq/Mouse/ChIPseqDataQuille2011.xls", sheet = "common genes")%>%as.data.frame  
  
  
chipSeqDataCleaner<-function(x){  
splitColoumnMinus<-data.frame(do.call('rbind', strsplit(as.character(x$location),'-',fixed=TRUE)))  
colnames(splitColoumnMinus)<- c("X1", "end")#re naming the coloumns  
splitColoumnSemiColon<-data.frame(do.call('rbind', strsplit(as.character(splitColoumnMinus$X1),':',fixed=TRUE)))  
colnames(splitColoumnSemiColon)<- c("chromosome", "start")#renaming those two  
geneSymbolMetaDataFromOriginalData<-x[2:3]  
dataFrameOfChipSeqData<- cbind(geneSymbolMetaDataFromOriginalData, splitColoumnSemiColon, splitColoumnMinus[2])%>%na.omit()  
removingTheNegatives<- cbind(dataFrameOfChipSeqData, (as.data.frame(as.numeric(as.character(dataFrameOfChipSeqData$end)))-as.data.frame(as.numeric(as.character(dataFrameOfChipSeqData$start)))))  
  
  
negativesRemoved<-subset(removingTheNegatives, removingTheNegatives$`as.numeric(as.character(dataFrameOfChipSeqData$end))`>0)  
grangeChipSeq<-makeGRangesFromDataFrame(negativesRemoved,  
 keep.extra.columns=FALSE,  
 ignore.strand=FALSE,  
 seqinfo=NULL,  
 seqnames.field=c("seqnames", "seqname",  
 "chromosome", "chrom",  
 "chr", "chromosome\_name",  
 "seqid"),  
 start.field="start",  
 end.field=c("end", "stop"),  
 strand.field="strand",  
 starts.in.df.are.0based=FALSE)  
}  
  
  
## Granges numbers  
grangeN2aChipSeq<-chipSeqDataCleaner(n2aChipSeq)  
grangeBrainChipSeq<-chipSeqDataCleaner(embyroChipSeq)  
grangeCommonChipSeq<-chipSeqDataCleaner(commonChipSeq)  
  
logical6Mer<-findOverlaps(arx6MerTFBS, grangeN2aChipSeq)%>%countLnodeHits()  
logicalJolma<-findOverlaps(arxJolmaTFBS, grangeN2aChipSeq)%>%countLnodeHits()  
logical4Mer<-findOverlaps(arx4MerTFBS, grangeN2aChipSeq)%>%countLnodeHits()  
overlap6Mer<- subset(arx6MerTFBS, logical6Mer)  
overlap4mer<- subset(arx4MerTFBS, logical4Mer)  
overlapJomal<- subset(arxJolmaTFBS, logicalJolma)  
  
arx6merDistance<-distanceToNearest(arx6MerTFBS, gtfUCSCgenes)%>%as.data.frame  
arx6merDistance<-arx6merDistance[3]  
  
  
ggplot(arx6merDistance, aes(x=distance))+  
 geom\_histogram(bins=1000, fill="#00b300")+  
 xlab(label = "Distance To The Closest Transcription Start Site(Base Pairs)")+  
 ylab(label= "Number of Motifs")+  
 theme(text = element\_text(size=12))+  
 scale\_x\_continuous(limits = c(0, 200000))+  
 scale\_y\_continuous(limits = c(0, 3000))

Arx 6mer and Jolma models are imported from the motifDB database, which is a compliment of many transcription factor motifs from different experiments. The 4mer i could not find in a database hence i developed the 4mer matrix myself. I then matched each of these to the mm9 genome which returned seperate GRange objects. Overlapping these objects with every genes, exons and enhancers from USCS and phantom 5 respectively. I think complied all of this into a table.

After this i took the chip confirmed Arx sites and overlapped them with arx motifs to check for the motif. To my surprise roughly 45% of these peaks actually contained Arx peaks hence bringing motif validity into question. As genes which contain the Arx motif are strongly enriched for the 6mer model it is evident that the motif is valid however, something else must be going on weather it is cofactors, other transcription factors or Arx can bind to multiple vastly different motifs. These chip-confirmed sites are sites reported in the 2011 Quile et al paper presented as a paper.