AprilW2

Awais Choudhry

10/04/2017

## 10-4-2017

The aim of the codce chunk below is to select for open chromatin containing an Arx TFBS hence indicating potentially active motifs in the brain. As the DNase seq track is cell type specific, different motifs would be expected to be active in different cell types. As Arx has approximately 2.1million potential binding sites simplely making some accessible and repressing the rest should be sufficent to regulate which genes arx can bind to for activation or repression. To do this i have taken a DNaseseq Track from Encode database by John Stamatoyannopoulos, UW.

###  
library(rtracklayer)  
library(magrittr)  
library(BSgenome.Mmusculus.UCSC.mm9)  
library(MotifDb)  
library(readxl)  
  
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)  
}  
  
  
  
grangeN2aChipSeq<-chipSeqDataCleaner(n2aChipSeq)  
grangeBrainChipSeq<-chipSeqDataCleaner(embyroChipSeq)  
grangeCommonChipSeq<-chipSeqDataCleaner(commonChipSeq)  
chipcConfirmedSites<- grangeCommonChipSeq  
  
##inputs for the for loop  
arx6Mer<-MotifDb::query(MotifDb, "Arx")[[6]]  
arx6Mer <- round(arx6Mer\*100)  
arx6MerTFBS<- matchPWM(arx6Mer, BSgenome.Mmusculus.UCSC.mm9, "90%", with.score = TRUE)  
transcriptionFactorModel<- arx6MerTFBS  
  
  
  
  
  
i<-1  
storage<-NULL  
for(i in 1:21){  
 chromosome<-c("chr1",  
 "chr2",   
 "chr3",  
 "chr4",  
 "chr5",  
 "chr6",  
 "chr7",  
 "chr8",  
 "chr9",  
 "chr10",  
 "chr11",  
 "chr12",  
 "chr13",  
 "chr14",  
 "chr15",  
 "chr16",  
 "chr17",  
 "chr18",  
 "chr19",  
 "chrX",  
 "chrY")[i]  
   
 dnaseSeqMouse <- import(con = "~/DataFiles/DNase/Mouse/ENCFF292LVM.bigWig", which = GRanges(chromosome, IRanges(1, end = .Machine$integer.max - 1)))  
 arx6chrM<-subset(transcriptionFactorModel, seqnames==chromosome)  
   
 seqlengths(dnaseSeqMouse)= seqlengths(arx6chrM)[names(seqlengths(dnaseSeqMouse))]  
 seqlevels(arx6chrM, force=TRUE)= seqlevels(dnaseSeqMouse)  
 intergers<-nearest(arx6chrM, dnaseSeqMouse)  
 grangesArxMotifWithDnaseSeq<-dnaseSeqMouse[intergers,]  
 tfbsWithScore<-cbind(arx6chrM, as.data.frame(grangesArxMotifWithDnaseSeq)[6])%>%na.omit()%>%makeGRangesFromDataFrame(  
 keep.extra.columns=TRUE,  
 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)  
   
   
 average2aDnaseI<-import(con = "~/DataFiles/DNase/Mouse/ENCFF292LVM.bigWig", which =chipcConfirmedSites)%>%as.data.frame()  
 averageN2a<-sum(average2aDnaseI[6])/dim(average2aDnaseI)[1]  
 potentialMotifs<-subset(tfbsWithScore, score.1>averageN2a)  
 storage<-rbind(storage, as.data.frame(potentialMotifs))  
 i<-i+1  
 print(storage)  
}  
grangeStorage<-makeGRangesFromDataFrame(storage, keep.extra.columns = TRUE)  
  
  
#ACtive Enhancers  
enhancerGrange <- import( con = "~/DataFiles/Enhancer Tracks/Mouse/Enhanceresmm9.bed")  
  
enhancerWithPotentialMotifs<-findOverlaps(enhancerGrange, grangeStorage)%>%countLnodeHits()  
ActiveArxEnhancers<-subset(enhancerGrange, enhancerWithPotentialMotifs)  
  
as.data.frame(ActiveArxEnhancers)%>%pander::pander()  
  
  
##seeing if there is an DnaseSeq SCore between and motif score!

Firstly we run some older code to process the ChIP-chip data for the brian, N2a cells and common genes. We import a chromosome of DNase seq data at a time due to the size of the track being too large for R to handle. Next we match the 6mer to the genome and attain a grange. Naturally we want to find the nearest, ideally 0, basepairs away DNase-Seq score. This will assign a value as to weather or not the motif is in trascriptionally active chromatin however, as the two objects differ in seq-levels we need to convert the DNase-seq levels into the UCSC seq levels so they are comptiable. Once do we can assign the closeset DNase-seq score to the Arx motif. Then subsetting for motifs in active chromatin and located in enhancers we attain 3 enhancers than contain at least 1 arx motif. These enhancers will undergo Hi-C analysis to see what they regulate.

11-4-2017

After reading about various methods to identify transcription factor motifs which are true as oppose to sites which occur by chance, i noticed they all take into account various epigenomic marks. Many algromatims take into account motif score, which is how accurate the motif is compared to the perfect motif and epigenomic environment of the moitf, hence how open the motif is. These two in conjuction allow individuals to test for Arx motifs.

##plotting to see if there is an inverse relationship between arx6mer TFBS score and DNaseSeq Score  
  
library(rtracklayer)  
library(GenomicRanges)  
library(MotifDb)  
library(BSgenome.Mmusculus.UCSC.mm9)  
library(readxl)  
library(ggplot2)  
library(Biostrings)  
  
arx6Mer<- MotifDb::query(MotifDb, "Arx")[[6]]  
arx6Mer<- round(arx6Mer\*100)  
arx6MerLow<- matchPWM(arx6Mer, BSgenome.Mmusculus.UCSC.mm9, "70%" )  
n2aChipSeq <- read\_excel("~/DataFiles/ChIPseq/Mouse/ChIPseqDataQuille2011.xls",   
 sheet = "only N2a")%>%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)  
}  
grangeN2aChipSeq<-chipSeqDataCleaner(n2aChipSeq)  
  
grangeMotifs<-subset(arx6MerLow, findOverlaps(grangeN2aChipSeq, arx6MerLow)%>%countRnodeHits())  
  
importDnaseGrange<-resize(grangeMotifs, 1000)  
dnaseSeqMouse <- import(con = "~/DataFiles/DNase/Mouse/ENCFF292LVM.bigWig", which = importDnaseGrange)  
  
seqlengths(dnaseSeqMouse)= seqlengths(arx6MerLow)[names(seqlengths(dnaseSeqMouse))]  
seqlevels(arx6MerLow, force=TRUE)= seqlevels(dnaseSeqMouse)  
intergers<-nearest( grangeMotifs, dnaseSeqMouse)  
ggplotDataFrame<-cbind(grangeMotifs, as.data.frame(values(dnaseSeqMouse[intergers,])))  
  
colnames(ggplotDataFrame)<- c("seqnames", "start", "end", "motif width",  
 "strand", "motif score", "DNA string", "DNase" )  
logGgplotDataFrame<-cbind(ggplotDataFrame, "logDnase"=log(as.data.frame(ggplotDataFrame$DNase)))  
colnames(logGgplotDataFrame)<- c("seqnames", "start", "end", "motif width","strand",  
 "motif score", "DNA string", "DNase", "logDnase" )  
  
ggplot(logGgplotDataFrame[logGgplotDataFrame$logDnase>2.5,], mapping = aes(x=`motif score`, y= DNase))+  
 geom\_smooth( )+  
 ggtitle("Relationship of Motif score to DNase score")+  
 theme\_bw()  
ggplot(logGgplotDataFrame, mapping = aes(x=`motif score`, y= DNase))+  
 geom\_point( )+  
 ggtitle("Motif Score vs DNase Score")+  
 theme\_bw()  
ggplot(logGgplotDataFrame, mapping = aes(x=`motif score`, y= logDnase))+  
 geom\_line( )+  
 ggtitle("Motif ")+  
 xlab("Motif Score")+  
 theme\_bw()  
ggplot(logGgplotDataFrame, mapping = aes( x= `motif score`))+   
 geom\_freqpoly()+  
 theme\_bw()  
ggplot(logGgplotDataFrame, mapping = aes(x=`motif score`, y= logDnase))+  
 geom\_histogram(stat="identity" )+  
 ggtitle("Motif ")+  
 xlab("Motif Score")+  
 theme\_bw()

To see if there was any relationship between Arx motif score and Dnase-seq track i imported the Dnase Seq track into R and matched Arx motifs to the genome however this time i did it with a low DNase seq score. After this i again attained the overlap of motif scores with Arx scores and converted the grange to a dataframe which will allow ggplot to read it. This inturn allowed me to plot motif score vs Dnase score on the x and y axis repsectively, allowing me to check for any relationship. I noticed the lower scoring motifs in open chromatin hence potentially illustrating a relationship, however, due to the sheer number of motifs in repressed chromatin this trend cannot be stastically signifincant. Further refinement and addition of histone modifications can improve the result.

12-4-2017

After another meeting with Tessa, as Cheryl is out for a few weeks i have was set with the task of generating a list of genes which contain Arx motifs that are confirmed by ChIP-chip. To do this was relatively simple in nature, i needed only to identify which ChIP sites have an Arx motif in them and then find out which gene they are closest to!

## boring table Script  
  
  
library(rtracklayer)  
library(GenomicRanges)  
library(MotifDb)  
library(BSgenome.Mmusculus.UCSC.mm9)  
library(readxl)  
library(ggplot2)  
  
arx6Mer<- query(MotifDb, "Arx")[[6]]  
arx6Mer<- round(arx6Mer\*100)  
arx6MerLow<- matchPWM(arx6Mer, BSgenome.Mmusculus.UCSC.mm9, "70%" )  
  
arxJolma<- query(MotifDb, "Arx")[[3]]  
arxJolma<- round(100\*arxJolma)  
arxJolmaLow<- matchPWM(arxJolma, BSgenome.Mmusculus.UCSC.mm9, "70%")  
  
##Importing the Data to see lengths  
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  
##Convert to Grange and clean  
  
  
chipSeqDataCleaner<- function(x){  
 library(magrittr)  
 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=TRUE,  
 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)  
}  
  
grangeN2aChipSeq<-chipSeqDataCleaner(n2aChipSeq)  
grangeBrainChipSeq<-chipSeqDataCleaner(embyroChipSeq)  
grangeCommonChipSeq<-chipSeqDataCleaner(commonChipSeq)  
  
##how many 6mers are in the cihp confirmed promoters  
  
intergerN2a6mer<-findOverlaps(arx6MerLow, grangeN2aChipSeq)%>%countRnodeHits()  
genesMotifArxN2a6mer<-length(subset(grangeN2aChipSeq, intergerN2a6mer))  
totalNumberGenesN2a<-length(grangeN2aChipSeq)  
  
intergerCommom6mer<-findOverlaps(arx6MerLow, grangeCommonChipSeq)%>%countRnodeHits()  
genesMotifArxCommomN2a6mer<-length(subset(grangeCommonChipSeq, intergerCommom6mer))  
totalNumberGenesCommon<-length(grangeCommonChipSeq)  
  
intergerBrain6mer<-findOverlaps(arx6MerLow, grangeBrainChipSeq)%>%countRnodeHits()  
genesMotifArxBrain6mer<-length(subset(grangeBrainChipSeq, intergerBrain6mer))  
totalNumberGenesBrain<-length(grangeBrainChipSeq)  
totalNumberArxMotifs6mer<-length(arx6MerLow)  
  
#how many Jolmas are in the chipconfirmed promoters!  
intergerBrainJolma<-findOverlaps(arxJolmaLow, grangeBrainChipSeq)%>%countRnodeHits()  
genesMotifArxBrainJolma<-length(subset(grangeBrainChipSeq, intergerBrainJolma))  
totalNumberGenesBrain<-length(grangeBrainChipSeq)  
  
intergerCommonJolma<-findOverlaps(arxJolmaLow, grangeCommonChipSeq)%>%countRnodeHits()  
genesMotifArxCommonJolma<-length(subset(grangeCommonChipSeq, intergerCommonJolma))  
totalNumberGenesCommon<-length(grangeCommonChipSeq)  
  
intergerN2aJolma<-findOverlaps(arxJolmaLow, grangeN2aChipSeq)%>%countRnodeHits()  
genesMotifArxN2aJolma<-length(subset(grangeN2aChipSeq, intergerN2aJolma))  
totalNumberGenesN2a<-length(grangeN2aChipSeq)  
totalNumberArxMotifsJolma<-length(arxJolmaLow)  
  
####GeneLists Note: it ris an addon to the 12/4/2017 from 13/4/2017  
  
##6mer  
listOfBrain6Mer<-as.data.frame(subset(grangeBrainChipSeq, intergerBrain6mer))[6]  
listOfN2a6Mer<-as.data.frame(subset(grangeN2aChipSeq, intergerN2a6mer))[6]  
listOfCommon6Mer<-as.data.frame(subset(grangeCommonChipSeq, intergerCommom6mer))[6]%>%as.vector  
capture.output(list(listOfCommon6Mer, listOfN2a6Mer, listOfBrain6Mer), file= "6merArxGenes.txt")  
##Jolma  
listOfBrainJolma<-as.data.frame(subset(grangeCommonChipSeq, intergerBrain6mer))[6]  
listOfN2aJolma<-as.data.frame(subset(grangeN2aChipSeq, intergerN2aJolma))[6]  
listOfCommonJolma<-as.data.frame(subset(grangeCommonChipSeq, intergerCommonJolma))[6]  
capture.output(list(listOfBrainJolma, listOfN2aJolma, listOfCommonJolma), file= "JolmaArxGenes.txt")

Firstly i imported the Arx chip-confirmed sites into R again and i generated a TFBS of 6mer. I then overlapped them and selected for motifs with the Arx motif in them. I imported the UCSC mm9 gene track from a bed file from the UCSC website. Ultising mm9 gene function i can find the nearest gene to the Arx motif and hence add it the GRange. I Then conver this to a dataframe and export the name of the gene ids as a text file which i sent to Cheryl.

16-4-2017 Below is some code i wrote to discover the relationship between Arx motif orientations: plaindormic or tandem, and spacing between motifs. The idea again here is that at a paritcular spacing and orientation we would expect to see more of a certain Arx motif as there would be selective pressure on these sites to be conserved. To investigate this i will be taking the 6mer motif and placing it in tandem and orientation and examining total counts.

## Generation for PWM for Planidromic sequences  
library(seqLogo)  
library(magrittr)  
library(GenomicRanges)  
library(ggplot2)  
library(magrittr)  
library(tibble)  
library(pander)  
library(reshape2)  
library(plyr)  
library(MotifDb)  
library(BSgenome.Mmusculus.UCSC.mm9)  
library(JASPAR2016)  
enhancerGrange <- import( con = "~/DataFiles/Enhancer Tracks/Mouse/Enhanceresmm9.bed")  
UCSCgenes<- import("~/Scripts/March/mm9.bed")  
promoters<- promoters(UCSCgenes)  
  
arx6MerPWM1space<-rbind( A=c(0,1,1,0,0,1,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,1,0,0,1,1,0))  
  
arx6MerPWM2space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0.25),  
 G=c(0,0,0,0,0,0,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,1,0,0,1,1,0))  
  
arx6MerPWM3space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,1,0,0,1,1,0))  
  
arx6MerPWM4space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
  
arx6MerPWM5space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
arx6MerPWM6space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
arx6MerPWM7space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
  
grangeplaindromic1space<-matchPWM(arx6MerPWM1space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic2space<-matchPWM(arx6MerPWM2space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic3space<-matchPWM(arx6MerPWM3space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic4space<-matchPWM(arx6MerPWM4space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic5space<-matchPWM(arx6MerPWM4space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic6space<-matchPWM(arx6MerPWM4space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeplaindromic7space<-matchPWM(arx6MerPWM4space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
  
##Databale results  
  
planindromicDataTable<-rbind(cbind(  
length(grangeplaindromic1space),  
Arx6mer<-sum(countOverlaps(grangeplaindromic1space, UCSCgenes)),  
sum(countOverlaps(grangeplaindromic1space, promoters)),  
sum(countOverlaps(grangeplaindromic1space, enhancerGrange))),  
   
cbind(length(grangeplaindromic2space),  
sum(countOverlaps(grangeplaindromic2space, UCSCgenes)),  
sum(countOverlaps(grangeplaindromic2space, promoters)),  
sum(countOverlaps(grangeplaindromic2space, enhancerGrange)))  
,  
cbind(  
numberOfArxSitesPlaindromic3Space<-length(grangeplaindromic3space),  
sum(countOverlaps(grangeplaindromic3space, UCSCgenes)),  
sum(countOverlaps(grangeplaindromic3space, promoters)),  
sum(countOverlaps(grangeplaindromic4space, enhancerGrange))),  
  
cbind(  
numberOfArxSitesPlaindromic4Space<-length(grangeplaindromic4space),  
sum(countOverlaps(grangeplaindromic4space, UCSCgenes)),  
sum(countOverlaps(grangeplaindromic4space, promoters)),  
sum(countOverlaps(grangeplaindromic4space, enhancerGrange)))  
)%>%as.data.frame()  
  
colnames(planindromicDataTable)<- c("Total",  
 "Motifs in genes",  
 "Motifs in Promoters",  
 "Motifs in Enhancers")  
rownames(planindromicDataTable)<- c("1 Space",  
 "2 Space",  
 "3 Space",  
 "4 Space")  
planindromicDataTable%>%pander()  
  
planindromicDataTable<- rownames\_to\_column(planindromicDataTable)  
  
ggplot(planindromicDataTable, aes(x=rowname, y=Total))+  
 geom\_bar(stat = "identity")+  
 theme\_bw()  
  
  
### Tandeom Sites  
  
arxTandem1Space<-rbind( A=c(0,1,1,0,0,1,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,1,0,0,1,1,0))  
arxTandem2Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,1,0,0,1,1,0))  
  
arxTandem3Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,1,0,0,1,1,0))  
arxTandem4Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
  
arxTandem5Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
arxTandem6Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
arxTandem7Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
arxTandem8Space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0.25,0.25,0.25,0) ,  
 T=c(1,0,0,1,1,0,0.25,0.25,0.25,0.25,1,0,0,1,1,0))  
  
grangeTandem1space<-matchPWM(arxTandem1Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem2space<-matchPWM(arxTandem2Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem3space<-matchPWM(arxTandem3Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem4space<-matchPWM(arxTandem4Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem5space<-matchPWM(arxTandem5Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem6space<-matchPWM(arxTandem6Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
grangeTandem7space<-matchPWM(arxTandem7Space, BSgenome.Mmusculus.UCSC.mm9, "90%")  
  
  
##Tandem DataTable  
tandemDataTable<-rbind(cbind(  
numberofTandem1spaceSites<-length(grangeTandem1space),  
dataTable1SpaceGenes<-sum(countOverlaps(grangeTandem1space, UCSCgenes)),  
dataTable1SpacePromoters<-sum(countOverlaps(grangeTandem1space, promoters)),  
dataTable1SpaceEnhancer<-sum(countOverlaps(grangeTandem1space, enhancerGrange))  
), cbind(  
numberofTandem2spaceSites<-length(grangeTandem2space),  
dataTable2SpaceGenes<-sum(countOverlaps(grangeTandem2space, UCSCgenes)),  
dataTable2SpacePromoters<-sum(countOverlaps(grangeTandem2space, promoters)),  
dataTable2SpaceEnhancer<-sum(countOverlaps(grangeTandem2space, enhancerGrange))  
), cbind(  
numberofTandem3spaceSites<-length(grangeTandem3space),  
dataTable3SpaceGenes<-sum(countOverlaps(grangeTandem3space, UCSCgenes)),  
dataTable3SpacePromoters<-sum(countOverlaps(grangeTandem3space, promoters)),  
dataTable3SpaceEnhancer<-sum(countOverlaps(grangeTandem3space, enhancerGrange))  
), cbind(  
numberofTandem4spaceSites<-length(grangeTandem4space),  
dataTable4SpaceGenes<-sum(countOverlaps(grangeTandem4space, UCSCgenes)),  
dataTable4SpacePromoters<-sum(countOverlaps(grangeTandem4space, promoters)),  
dataTable4SpaceEnhancer<-sum(countOverlaps(grangeTandem4space, enhancerGrange))  
)  
)%>%as.data.frame  
  
colnames(tandemDataTable)<- c("Total",   
 "Motifs in genes",  
 "Motifs in promoters",  
 "Motifs in enhancers")  
rownames(tandemDataTable)<-c("1 Space",  
 "2 Space",  
 "3 Space",  
 "4 Space")  
tandemDataTable%>%pander()  
  
tandemDataTable<-rownames\_to\_column(tandemDataTable)  
  
ggplot(tandemDataTable, aes(x= rowname, y= Total))+  
 geom\_bar(stat="identity")+  
 theme\_bw()

The first thing i did is i generate position weight matrices for each orientation of Arx, tandem and plaindromic. Then for each orientation i spaced the motifs out by 1 additional base going all the way up to 7. In addition, i did a minus one base to see if Arx motifs prefer to be closer together as suggest by Jolma in his CAP-SELEX. After doing so i match each to the genome and mapped these to the genome and counted the total numbers. Any orientations or spacing of interest should have more motifs than the others, as they would be there only by chance. From this analysis at 90% we see Arx tandem at 2 spacing being higher than arx the others hence prehaps being on some signficance. In addition, we see in the plaindromic orientation that the 4 spacing maybe of relevance.

In addition, i took the first 4 spacings for each orientation and plotted a histogram to see where these motifs are in relation to the closest promoter region. As expected we can see an exponential decay of number of motifs with increasing distance away from promoters. All spacings show similar trends simply at different counts.