June W4

Awais Choudhry

2017-6-27

### June W4

library(magrittr)  
library(rtracklayer)

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':  
##   
## clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
## clusterExport, clusterMap, parApply, parCapply, parLapply,  
## parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':  
##   
## IQR, mad, xtabs

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, append, as.data.frame, cbind, colnames,  
## do.call, duplicated, eval, evalq, Filter, Find, get, grep,  
## grepl, intersect, is.unsorted, lapply, lengths, Map, mapply,  
## match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,  
## Position, rank, rbind, Reduce, rownames, sapply, setdiff,  
## sort, table, tapply, union, unique, unsplit, which, which.max,  
## which.min

## Loading required package: S4Vectors

##   
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':  
##   
## colMeans, colSums, expand.grid, rowMeans, rowSums

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

library(GenomicRanges)  
library(Biostrings)

## Loading required package: XVector

library(BSgenome.Hsapiens.UCSC.hg19)

## Loading required package: BSgenome

#arx6merTFBS<-matchPWM( round(PWM("TAATTA")\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
chromHMMHippocampus<-import("~/DataFiles/ChromHMM/human/E071\_15\_coreMarks\_dense.bed.gz")

Before i imported all this data I used the netToBed UCSC genome tool which caused me a great deal of stress due to permissions.

I was unable to execute it until i found out that i needed to change the permissions which i was again unable to do until i changed the format of the harddrive in which it was executing! Once i figured that out, it was fairly straight foward to execute, i have uploaded the .bed file to my VM where i will examine which motifs are found in conserved regions of human-mouse genome

humanMouseConservedRegions<-import("~/DataFiles/Conservation/Human/h19.mm9.bed")  
ConservedMotifsBetweenHumanMouse<-subset(arx6merTFBS, findOverlaps(arx6merTFBS, humanMouseConservedRegions, minoverlap = 6)%>%countLnodeHits())  
  
ConservedMotifsBetweenHumanMouse

## GRanges object with 1028250 ranges and 2 metadata columns:  
## seqnames ranges strand | score  
## <Rle> <IRanges> <Rle> | <numeric>  
## [1] chr1 [16368, 16373] + | 6  
## [2] chr1 [25599, 25604] + | 6  
## [3] chr1 [52597, 52602] + | 6  
## [4] chr1 [57524, 57529] + | 6  
## [5] chr1 [57670, 57675] + | 6  
## ... ... ... ... . ...  
## [1028246] chrUn\_gl000249 [36259, 36264] + | 6  
## [1028247] chrUn\_gl000249 [37925, 37930] + | 6  
## [1028248] chrUn\_gl000249 [32191, 32196] - | 6  
## [1028249] chrUn\_gl000249 [36259, 36264] - | 6  
## [1028250] chrUn\_gl000249 [37925, 37930] - | 6  
## string  
## <DNAStringSet>  
## [1] TAATTA  
## [2] TAATTA  
## [3] TAATTA  
## [4] TAATTA  
## [5] TAATTA  
## ... ...  
## [1028246] TAATTA  
## [1028247] TAATTA  
## [1028248] TAATTA  
## [1028249] TAATTA  
## [1028250] TAATTA  
## -------  
## seqinfo: 93 sequences from an unspecified genome

There are 2.8million motifs in total in the human genome where roughly 1 million of these motifs fall into where these motifs are conserved. This, technique alone does not sufficently reduce the number of moitfs down to indicate which of these motifs are functional/indicative of where ARX binds.

## Graph of illustrating the number of these motifs.

To execute this code i require code from June W3, i placed it in the code chunk below.

library(Biostrings)  
library(BSgenome.Hsapiens.UCSC.hg19)  
library(magrittr)  
library(tibble)  
  
##Get all ARX Binding sites in humans  
# arx6merTFBS<-matchPWM(round((PWM("TAATTA"))\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
  
##Get all the Conservation Scores for these regions  
phyloPscores<-import("~/DataFiles/Conservation/Human/hg19.100way.phyloP100way.bw",which =arx6merTFBS)

## Warning in .local(con, format, text, ...): 'which' contains seqlevels not  
## known to BigWig file: chrUn\_gl000226

##Subset for only the most conserved regions  
#we do a 1.3 instead of 0.9 as phylop is scores between -14 and 3 hence, 90% conservation is 15.3  
phyloPTrack<-subset(phyloPscores, score>=1.3)  
  
##See which Motifs fall into these regions  
polyPConserved<-subset(arx6merTFBS,findOverlaps(phyloPTrack, arx6merTFBS)%>%countRnodeHits())  
  
##Phast Con Scores Same as above  
phastConScores<-import("~/DataFiles/Conservation/Human/hg19.100way.phastCons.bw",which =arx6merTFBS)

## Warning in .local(con, format, text, ...): 'which' contains seqlevels not  
## known to BigWig file: chrUn\_gl000226

PhastConTrack<-subset(phastConScores, score>=0.9)  
  
##See which Motifs fall into these regions  
phastConserved<-subset(arx6merTFBS,findOverlaps(PhastConTrack, arx6merTFBS)%>%countRnodeHits())

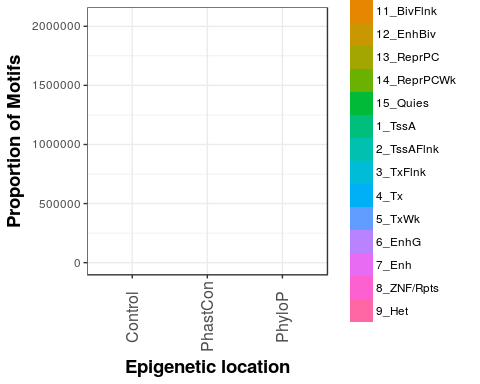
Additionally, i would like to compare where these motifs fall inrelation to chromHMM tracks, i am using the Hippocampus chromHMM track to identify where these motifs fall. Due to transcription factors being tissue specific this would provide a accurate result.

chromHMMHippocampus<-import("~/DataFiles/ChromHMM/human/E071\_15\_coreMarks\_dense.bed.gz")  
  
  
#List Of all Unique sites  
uniqueRegions<-as.list(unique(mcols(chromHMMHippocampus)$name))  
  
##Control/All ARX motifs  
##Phylop Motif Locations  
hippocampusLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), arx6merTFBS))}  
HippoCampusNumberOfMotifsControl<-lapply(X = uniqueRegions, FUN = hippocampusLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifsControl)<-uniqueRegions  
  
  
  
##PhastCon Motif Locations  
hippocampusLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), phastConserved))}  
HippoCampusNumberOfMotifsPhastCon<-lapply(X = uniqueRegions, FUN = hippocampusLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifsPhastCon)<-uniqueRegions  
  
##Phylop Motif Locations  
hippocampusLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), phyloPTrack))}  
HippoCampusNumberOfMotifsPhyloP<-lapply(X = uniqueRegions, FUN = hippocampusLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifsPhyloP)<-uniqueRegions  
  
  
  
library(reshape)

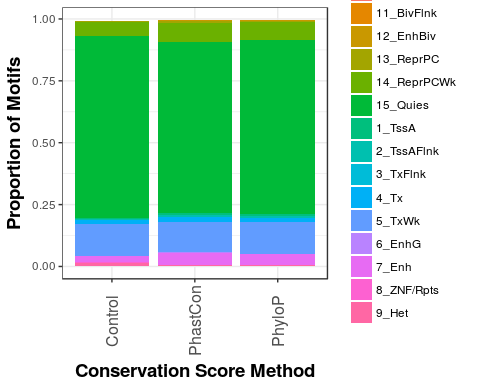
##   
## Attaching package: 'reshape'

## The following objects are masked from 'package:S4Vectors':  
##   
## expand, rename

library(ggplot2)  
  
HippoCampusNumberOfMotifsPhyloP<-t(HippoCampusNumberOfMotifsPhyloP)%>%as.data.frame()%>%rownames\_to\_column()  
HippoCampusNumberOfMotifsPhastCon<-t(HippoCampusNumberOfMotifsPhastCon)%>%as.data.frame()%>%rownames\_to\_column()  
HippoCampusNumberOfMotifsControl<-t(HippoCampusNumberOfMotifsControl)%>%as.data.frame()%>%rownames\_to\_column()  
  
   
 allConservationMotifMethodLocations<-cbind(HippoCampusNumberOfMotifsControl, HippoCampusNumberOfMotifsPhastCon[2],HippoCampusNumberOfMotifsPhyloP[2] )  
 colnames(allConservationMotifMethodLocations)<-c("Location", "Control", "PhastCon", "PhyloP")  
   
 reshapedHippocampusConservation<-melt(allConservationMotifMethodLocations, id.vars= c("Location"))  
   
ggplot(reshapedHippocampusConservation, aes(x=`variable`, y= `value`))+  
 geom\_tile(aes(fill= `Location`), stat= "identity")+  
 xlab(label= "Epigenetic location")+  
 ylab(label= "Proportion of Motifs")+  
 guides(fill=guide\_legend(title="Genomic Location"))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, angle = 90, vjust = 0.5),  
 axis.title=element\_text(size=14,face="bold"))



## Normalising by dividing by the total number of motifs. It is clear that ARX is highly expressed.   
proportionPhyloP<-HippoCampusNumberOfMotifsPhyloP  
proprtionPhastCon<-HippoCampusNumberOfMotifsPhastCon  
proportionControl<-HippoCampusNumberOfMotifsControl  
  
proportionPhyloP$V1<-proportionPhyloP$V1/length(phyloPTrack)  
proprtionPhastCon$V1<-proprtionPhastCon$V1/length(phastConserved)  
proportionControl$V1<-proportionControl$V1/length(arx6merTFBS)  
  
 allConservationMotifMethodLocationsProportions<-cbind(proportionControl, proprtionPhastCon[2],proportionPhyloP[2] )  
 colnames(allConservationMotifMethodLocationsProportions)<-c("Location", "Control", "PhastCon", "PhyloP")  
   
 reshapedHippocampusConservationProprtion<-melt(allConservationMotifMethodLocationsProportions, id.vars= c("Location"))  
   
ggplot(reshapedHippocampusConservationProprtion, aes(x=`variable`, y= `value`))+  
 geom\_bar(aes(fill= `Location`), stat= "identity")+  
 xlab(label= "Conservation Score Method")+  
 ylab(label= "Proportion of Motifs")+  
 guides(fill=guide\_legend(title="Genomic Location"))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, angle = 90, vjust = 0.5),  
 axis.title=element\_text(size=14,face="bold"))

 This graph plots protion of motifs and total number in each chromatin state in the human hippocampus tissue. Evidently it is clear that PhastCon method provides the most stringent method for selecting for motifs. Secondly, all conservational models show little to no-enrichement in enhancers, or promoters. This can be readily attributed to the AATT core being a componmenet of a number of transcription factors hence these motifs are readily bound to by multiple transcription factors which may not be active in this tissue type. In addition, because ARX is regulated in many different tissues, with little overlap in the genes it regulates between tissues we would expect to observe this trend.

29/6/2017 # Identifying which of these 2 spaced motifs holds up/ is conserved more?

Here we are simply comparing the proportions of Arx motif models.

As preivous papers found longer motifs typically predict for TFBS better hence we wanted to see if a longer version of ARX's motif would better predict for TFBS.

library(Biostrings)  
library(magrittr)  
library(GenomicRanges)  
library(rtracklayer)  
library(BSgenome.Hsapiens.UCSC.hg19)  
library(tibble)  
  
arxTandem2Spaced<-cbind(round(PWM("TAATTA")\*7), 0.25, 0.25, round(PWM("TAATTA")\*7))  
  
arxTandem2SpacedTFBS<-matchPWM(arxTandem2Spaced, BSgenome.Hsapiens.UCSC.hg19, "100%")  
  
conservedArxTandem2SpacedTFBS<-phastConserved<-subset(arxTandem2SpacedTFBS,findOverlaps(PhastConTrack, arxTandem2SpacedTFBS)%>%countRnodeHits())  
  
  
normalisedProportion<-cbind(length(conservedArxTandem2SpacedTFBS)/length(arxTandem2SpacedTFBS),   
 length(phastConserved)/ length(arx6merTFBS))%>%as.data.frame()  
colnames(normalisedProportion)<-c("Tandem 2 Spaced", "Control")  
normalisedProportion<-t(normalisedProportion)  
colnames(normalisedProportion)<-c("Proportion Of Motifs")  
normalisedProportion<- rownames\_to\_column(as.data.frame(normalisedProportion))  
normalisedProportion<-as.data.frame(normalisedProportion)  
  
library(ggplot2)  
  
ggplot(normalisedProportion, aes(x=rowname, y= `Proportion Of Motifs`))+  
 geom\_bar(stat="identity", aes(fill=rowname))+  
 xlab(label= "Motif Model")+  
 ylab(label= "Proportion of Motifs")+  
 guides(fill=guide\_legend(title="Motif Model"))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, angle = 90, vjust = 0.5),  
 axis.title=element\_text(size=14,face="bold"))

From this Code chunk we can see that a significnatly larger proportion of arx6mer motifs falling into conserved regions than Tandem 2 spaced motifs. This suggests that the 2spaced 6mer motif maybe not functional, or gained function much more recently. This in conjuction with much of the support form other papers suggests that conservation scores are not the ideal way to identify functional motifs however, this method does drastically reduce false positives.

In conculsion, it is clear, that utilising phylogenetic approaches to motif binding it is not sufficent hence we will use cis-regulatory modules.

## Cis regulatory Module approaches

Really all i am doing is overlapping it with the motifs but there are so few annotated in this database it will be easier just to use the enhancer regions as they are essentially cis-regualtory modules.

##import every gene expressed in the brain  
##importing the CIS regulatory modules  
library(readr)  
library(magrittr)  
library(Gviz)  
library(rtracklayer)  
CRMdataframe<-read\_delim("~/DataFiles/Cis Regulatory Module/Human/crm\_inf.txt",   
 "\t", escape\_double = FALSE, trim\_ws = TRUE)  
  
  
colnames(CRMdataframe)<-c("chrom", "start", "end" , "RefSeq", "TSS", "Orientation", "Tissue", "Min Energy", "Transcription Factors")  
crmGrange<-CRMdataframe%>%makeGRangesFromDataFrame(  
 keep.extra.columns=TRUE,  
 ignore.strand=FALSE,  
 seqinfo=NULL,  
 seqnames.field=c("seqnames", "seqname",  
 "chromosome", "chrom",  
 "chr", "chromosome\_name",  
 "seqid"))  
  
extendedCRMGrange<-crmGrange+100  
  
NumberOfMotifsInCRMs<-subset(extendedCRMGrange, findOverlaps(extendedCRMGrange, arx6merTFBS)%>%countLnodeHits())

# Identifying which transcription factors are commonly found with ARX genes

library(readr)  
bed <- read\_delim("~/DataFiles/Quille/bed",   
 "\t", escape\_double = FALSE, trim\_ws = TRUE)  
  
ARXgenes<-unique(bed$mm10.kgXref.geneSymbol)%>%as.data.frame()  
  
  
write.csv(ARXgenes, file= "ARXgenes.bed", sep = "\t")

We conducted an enirchment using this gene list using the web based tool EnrichR. The results were not particularly interesting. so thats no good. Arx was not enriched, however, this is likely due to these databases typically only taking high-quality ChiP-seq motifs.

## Purely ARX modules

The question i am asking here is how many ARX motifs can be found 500bp up stream and 500bp down stream. Secondly i want to know where these motif clusters are found. - knowing they are not all active in one tissue type. Thirdly: What is the conservation score of these motifs? - knowing they are likely not always active

library(Biostrings)  
library(BSgenome.Hsapiens.UCSC.hg19)  
numberOfMotifsInArxCrm<-8  
  
  
#arx6merTFBS<-matchPWM( round(PWM("TAATTA")\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
  
arx1kb<- arx6merTFBS+500  
  
clustersOfARXSpecificTFBS<-subset(arx6merTFBS, countOverlaps(arx1kb, arx6merTFBS)>=numberOfMotifsInArxCrm)  
  
library(rtracklayer)  
library(ggplot2)  
  
#PHastCon scores from previous chunnks  
clustersConserved<-subset(clustersOfARXSpecificTFBS,findOverlaps(PhastConTrack, clustersOfARXSpecificTFBS)%>%countRnodeHits())  
  
proportionOfConservedClusters<-cbind(length(conservedArxTandem2SpacedTFBS)/length(arxTandem2SpacedTFBS),   
 length(phastConserved)/ length(arx6merTFBS),  
 length(clustersConserved)/length(clustersOfARXSpecificTFBS))%>%as.data.frame()  
  
colnames(proportionOfConservedClusters)<-c("Tandem 2 Spaced Conserved", "ARX 6mer", "Clustered Sites")  
  
proportionOfConservedClusters<-t(proportionOfConservedClusters)%>%as.data.frame()  
  
proportionOfConservedClusters<- rownames\_to\_column(proportionOfConservedClusters, var= "Motif Model")%>%as.data.frame  
  
  
  
  
ggplot(proportionOfConservedClusters, aes(x=`Motif Model`, y=V1))+  
 geom\_bar(stat="identity", aes(fill= `Motif Model`))+  
 theme\_bw()+   
 xlab(label= "Motif Model")+  
 ylab(label= "Proportion of Motifs")+  
 guides(fill=guide\_legend(title="Motif Model"))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, vjust = 0.5),  
 axis.title=element\_text(size=14,face="bold"))

So these clusters again are not particularly conserved probably because they all fall into simple tandem repeats that are non functional.

# Okay well Where are these CLUSTERS located?

To identify this i will be systematically identifying where each group of these elements is found by using a chromHMM track.

I fully anticipate a significant deal to not be in functionally active chromatin due to ARX acting so many many cell types.

library(rtracklayer)  
  
chromHMMHippocampus<-import("~/DataFiles/ChromHMM/human/E071\_15\_coreMarks\_dense.bed.gz")  
  
#List Of all Unique sites  
uniqueRegions<-as.list(unique(mcols(chromHMMHippocampus)$name))  
  
##Control/All ARX motifs  
##Phylop Motif Locations  
hippocampusClusterLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), clustersConserved))}  
HippoCampusNumberOfMotifsControl<-lapply(X = uniqueRegions, FUN = hippocampusClusterLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifsControl)<-uniqueRegions  
  
HippoCampusNumberOfMotifsControl<-t(HippoCampusNumberOfMotifsControl)%>%as.data.frame()%>%rownames\_to\_column()  
  
ggplot(HippoCampusNumberOfMotifsControl, aes(x=rowname, y= V1))+  
 geom\_bar(stat="identity", aes(fill=rowname))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, vjust = -0.00,angle= 90),  
 axis.title=element\_text(size=14,face="bold"))+  
 xlab(label= "Epigenetic location")+  
 ylab(label= "Number of Motifs")+  
 guides(fill=guide\_legend(title="Genomic Location"))

Interesentingly we do see a slight enrichement in the brain ish region? Maybe i am just seeing things because i want it.

Also these colors suck. we need to re-do the colours

## identfying where these Arx motifs are located!

Just out of our curiosity we want to idenitfy which transponsable elements are generating most of our motifs as in some instances transponsable elements can code for enhancer regions as they contian clusters of TFBS. Hence we wanted to idnetify which genomic regions

transponsableElements<-import("~/DataFiles/Transposable elements/Human/transponsableElement.bed")  
  
uniqueTransponsableElements<-as.list(unique(mcols(transponsableElements)$name))  
  
transponsableElementsFunction<-function(x){length(findOverlaps(subset(transponsableElements, name==x), arx6merTFBS))}  
  
library(parallel)  
  
mostTheConservation<-lapply(uniqueTransponsableElements, transponsableElementsFunction)  
mostTheConservation<-mostTheConservation%>%as.matrix()  
uniqueTransponsableElements<-uniqueTransponsableElements%>%as.matrix()  
colnamesTransponsable<-cbind(uniqueTransponsableElements, mostTheConservation)%>%as.data.frame()  
colnames(colnamesTransponsable)<- c("Transponsable", "Number Of Motifs")  
colnamesTransponsable$`Number Of Motifs`<-colnamesTransponsable$`Number Of Motifs`%>%as.numeric()  
colnamesTransponsable$Transponsable<-colnamesTransponsable$Transponsable%>%as.character()  
  
mostFrequentElements<-subset(colnamesTransponsable, `Number Of Motifs`>=7000)  
  
library(ggplot2)  
  
ggplot(mostFrequentElements, aes(x=Transponsable, y= `Number Of Motifs`))+  
 geom\_bar( stat="identity", aes(fill= `Transponsable`))+  
 theme\_bw()+  
 theme(axis.text.x=element\_text(size=12, vjust = -0.00,angle= 90),  
 axis.title=element\_text(size=14,face="bold"))+  
 xlab(label= "Transponsable Element")+  
 ylab(label= "Number of Motifs")+  
 guides(fill=guide\_legend(title="Transponsable Element"))