June W3

Awais Choudhry

2017-6-22

22/6/2017

## Cis Regulatory Modules

I have donwloaded a cis regulatory module track from <http://bioinfo.wilmer.jhu.edu/tiger/> and i am going to see just how many ARX motifs are found. The terrific thing about this CRM model is that it is prebuilt on a selection of TFs for each tissue. However, I do have issues with it as it does not specificy which genome it is from and it is quite old (2008)

##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+500  
  
NumberOfMotifsInCRMs<-findOverlaps(extendedCRMGrange, arx6merTFBS)%>%length()

In a set of CRMS downloaded off and just did a quick overlap to find only 542 motifs in CRMS in all tissue types. This appears to be extremely low and has raised my suspicans. Henceforth, i will do this step manually by taking ChIP data for transcription factors expressed in specific tissues to develop CRMs. I will however, continue my search for a database where i can download confirmed, or literature supported Cis regulatory modules.

## 27/6/2017

I have searched aroung sidnificantly and it appears clustering of TFBS is a common method to identify CRMS.

I have lost the code however, to identify CIS regulatory regions i donwloaded bedfiles off of UCSC for TFBS and imported into r using rtracklayer. I then increased the size by 500 and overlapped it with itself counting the number of overlaps. Every region with 5 or more overlaps i classified as a CRM. I overlapped this with the database CRMs and got 10% overlapping. Hence i had 90% novel/False positive CRMs. Consequnece of Different cell types and TFBS i believe mine is likely wrong. Integration of mulitple other layers such as DNAase/ATAC-seq or even multiple histone modificaitons would help support this. In conculsion, sequnece information is insufficent.

## ChromHMM

Lets find out where ARX motifs are in each regon of the genome.

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(Gviz)

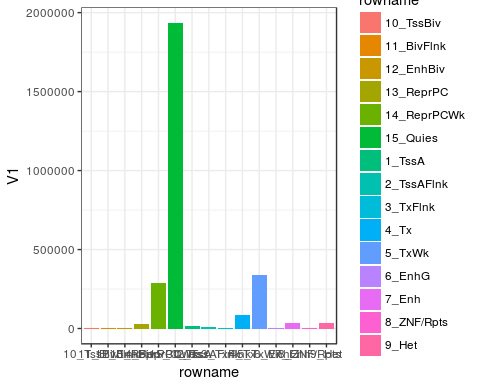
## Loading required package: grid

library(magrittr)  
library(tibble)  
  
  
  
  
### Importing in the chromHMM TRACKS  
chromHMMHippocampus<-import("~/DataFiles/ChromHMM/human/E071\_15\_coreMarks\_dense.bed.gz")  
chromHMMStomache<-import("~/DataFiles/ChromHMM/human/E111\_15\_coreMarks\_dense.bed.gz")  
  
  
  
#arx6merTFBS<-matchPWM(round((PWM("TAATTA"))\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
  
#List Of all Unique sites  
uniqueRegions<-as.list(unique(mcols(chromHMMHippocampus)$name))  
  
hippocampusLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), arx6merTFBS))}  
  
HippoCampusNumberOfMotifs<-lapply(X = uniqueRegions, FUN = hippocampusLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifs)<-uniqueRegions  
  
  
StomacheLocationFunction<-function(x){length(findOverlaps(subset(chromHMMStomache, name==x), arx6merTFBS))}  
  
stomacheNumberOfMotifs<-lapply(X = uniqueRegions, FUN = StomacheLocationFunction)%>%as.data.frame()  
colnames(stomacheNumberOfMotifs)<-uniqueRegions  
  
library(reshape)

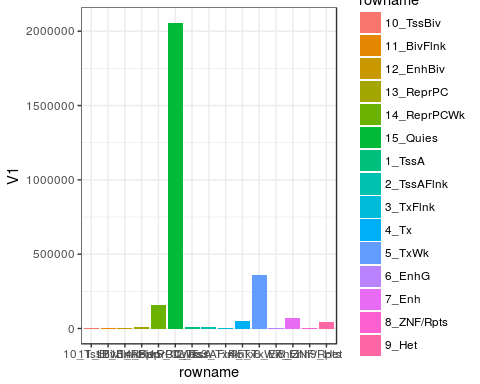
##   
## Attaching package: 'reshape'

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

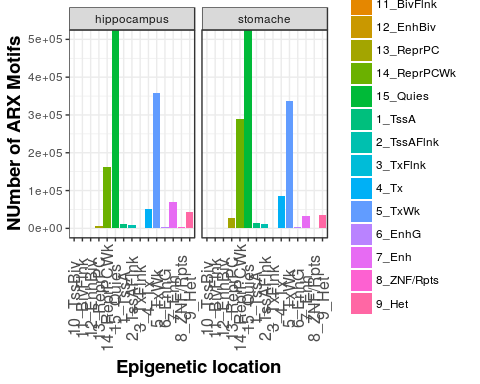
library(ggplot2)  
  
stomacheNumberOfMotifs<-t(stomacheNumberOfMotifs)%>%as.data.frame()%>%rownames\_to\_column()  
HippoCampusNumberOfMotifs<-t(HippoCampusNumberOfMotifs)%>%as.data.frame()%>%rownames\_to\_column()  
  
  
  
ggplot(stomacheNumberOfMotifs, aes(x=rowname, y= V1, fill=rowname))+  
 geom\_bar(stat="identity")+  
 theme\_bw()



ggplot(HippoCampusNumberOfMotifs, aes(x=rowname, y= V1, fill=rowname))+  
 geom\_bar(stat="identity")+  
 theme\_bw()



colnames(stomacheNumberOfMotifs)<-c("Stomache", "Stomache Motifs")  
 colnames(HippoCampusNumberOfMotifs)<-c("Hippocampus", "Hippocampus Motifs")  
   
 StomacheAndHippoCampus<-cbind(HippoCampusNumberOfMotifs, stomacheNumberOfMotifs)  
 reshapedHippocampusStomache<-reshape( data =StomacheAndHippoCampus ,  
 varying = c( "Hippocampus Motifs", "Stomache Motifs"),  
 v.names = "Numbers of Motif",  
 timevar = "Location",  
 times = c("hippocampus", "stomache"),  
 direction = "long")  
   
ggplot(reshapedHippocampusStomache, aes(x=`Stomache`, y= `Numbers of Motif`))+  
 geom\_bar(aes(fill= `Stomache`), stat= "identity")+  
 facet\_wrap(~Location)+  
 xlab(label= "Epigenetic location")+  
 ylab(label= "NUmber of ARX 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"))+  
 coord\_cartesian(ylim = c(0, 500000))



Here i have plotted the locations of ARX motifs based on epigenetic markers for specific cell lines/tissues. In this case it appears that the 6mer, despite not being specific enough is storngly enriched in ARX positive locations. This is further indicative that JUST the sequence alone is not indicative of where ARX binds. To rectify this, we all now examine the conserved motif scores.

## Phylogenetic scores

Identifiyng motifs in conserved regions

To do this, we import the conservation scores of genomic regions with Arx motifs and hard filter them for regions that score highly and then intersect them.

EDIT: Note we made the mistake of not enusruing the whole motif falls into the conserved region. Anyoverlap is required.

library(Biostrings)

## Loading required package: XVector

library(BSgenome.Hsapiens.UCSC.hg19)

## Loading required package: BSgenome

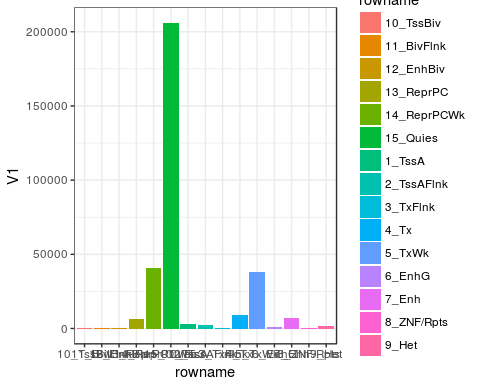
library(magrittr)  
library(tibble)  
  
##Get all ARX Binding sites in humans  
#arx6merTFBS<-matchPWM(round((PWM("TAATTA"))\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
##Get all the Conservation Scores for these regions  
phyloPscores<-import("~/DataFiles/Conservation/Human/hg19.100way.phyloP100way.bw",which =arx6merTFBS)  
  
##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)  
  
PhastConTrack<-subset(phastConScores, score>=0.9)  
  
##See which Motifs fall into these regions  
phastConserved<-subset(arx6merTFBS,findOverlaps(PhastConTrack, arx6merTFBS)%>%countRnodeHits())

This code chunk extracts the conservation scores (either phastcon, or Phyolp) from 100 way alignment. These scores are then subsetted for regions which score equal to or above 0.9. The ARX motifs that overlap or are found in these regions are then extracted and I have called them based on the method of which the conservation of the region is scored, ie phastConserved and phloPTrack. As phastCon appears to be more stringent for returning less motifs I will opt for this method.

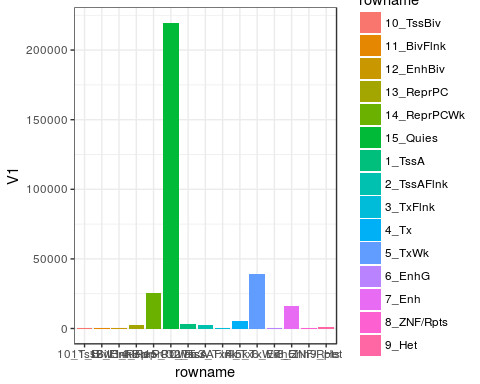
I do not fully understand how the difference in scoring between these methods apart from phyloP scores are based around the rate of evoltuion, hence positive intergers indicate the base is evolving at a rate slower than evolution whereas a negative indicates that the ARX motif is moving faster than evolution hence is more and less conserved respectively.

## HeatMap of Epigenomics Trail

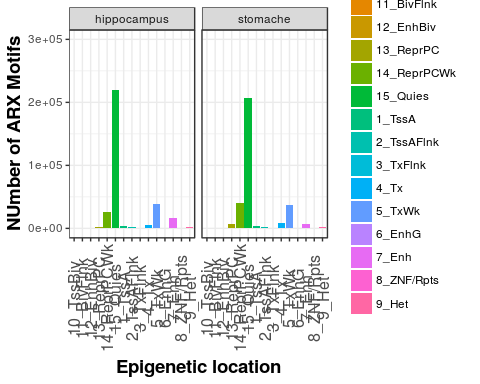
library(rtracklayer)  
library(Gviz)  
library(Biostrings)  
library(BSgenome.Hsapiens.UCSC.hg19)  
chromHMMHippocampus<-import("~/DataFiles/ChromHMM/human/E071\_15\_coreMarks\_dense.bed.gz")  
chromHMMStomache<-import("~/DataFiles/ChromHMM/human/E111\_15\_coreMarks\_dense.bed.gz")  
  
  
  
#arx6merTFBS<-matchPWM(round((PWM("TAATTA"))\*7), BSgenome.Hsapiens.UCSC.hg19, "100%")  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
arxMotifInput<-phastConserved  
  
#List Of all Unique sites  
uniqueRegions<-as.list(unique(mcols(chromHMMHippocampus)$name))  
  
hippocampusLocationFunction<-function(x){length(findOverlaps(subset(chromHMMHippocampus, name==x), arxMotifInput, minoverlap = 6))}  
  
HippoCampusNumberOfMotifs<-lapply(X = uniqueRegions, FUN = hippocampusLocationFunction)%>%as.data.frame()  
colnames(HippoCampusNumberOfMotifs)<-uniqueRegions  
  
  
StomacheLocationFunction<-function(x){length(findOverlaps(subset(chromHMMStomache, name==x), arxMotifInput, minoverlap = 6))}  
  
stomacheNumberOfMotifs<-lapply(X = uniqueRegions, FUN = StomacheLocationFunction)%>%as.data.frame()  
colnames(stomacheNumberOfMotifs)<-uniqueRegions  
  
library(reshape)  
library(ggplot2)  
  
stomacheNumberOfMotifs<-t(stomacheNumberOfMotifs)%>%as.data.frame()%>%rownames\_to\_column()  
HippoCampusNumberOfMotifs<-t(HippoCampusNumberOfMotifs)%>%as.data.frame()%>%rownames\_to\_column()  
  
  
  
ggplot(stomacheNumberOfMotifs, aes(x=rowname, y= V1, fill=rowname))+  
 geom\_bar(stat="identity")+  
 theme\_bw()



ggplot(HippoCampusNumberOfMotifs, aes(x=rowname, y= V1, fill=rowname))+  
 geom\_bar(stat="identity")+  
 theme\_bw()



colnames(stomacheNumberOfMotifs)<-c("Stomache", "Stomache Motifs")  
 colnames(HippoCampusNumberOfMotifs)<-c("Hippocampus", "Hippocampus Motifs")  
   
 StomacheAndHippoCampus<-cbind(HippoCampusNumberOfMotifs, stomacheNumberOfMotifs)  
 reshapedHippocampusStomache<-reshape( data =StomacheAndHippoCampus ,  
 varying = c( "Hippocampus Motifs", "Stomache Motifs"),  
 v.names = "Numbers of Motif",  
 timevar = "Location",  
 times = c("hippocampus", "stomache"),  
 direction = "long")  
   
ggplot(reshapedHippocampusStomache, aes(x=`Stomache`, y= `Numbers of Motif`))+  
 geom\_bar(aes(fill= `Stomache`), stat= "identity")+  
 facet\_wrap(~Location)+  
 xlab(label= "Epigenetic location")+  
 ylab(label= "NUmber of ARX 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"))+  
 coord\_cartesian(ylim = c(0, 300000))



A short comming of this method is that it only uses the overall conservation between 100 or so species. Upon further investigation it appears i need to download the alignments myself and then run phastCon /PhyloP on the data to attain just a human and mouse conservation score and human primate. In addition, this method appears to return far too many positive results, as it reduces the total from 2.5 million to 1.5 million approximately. I thinking the script is identifying all motifs that have at least 1 base that overlaps with a conserved region. My biggest issue with this is that if the entire motif is not conserved then it is not a conserved motif. Henceforth i need to find a method to tell the script to return me only motifs where every base in the motif has a conservation score greater than 0.9.

## Edit 26/6/2016: I have found a method to improve the phylogenetic results. It is a simple fix where you can instruct the function of findOverlaps to give a minimum overlap. I can set this to 100bp.