Human Genes Interacting List

Awais Choudhry

09/10/2017

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(magrittr)  
library(BSgenome.Hsapiens.UCSC.hg19)

## Loading required package: BSgenome

## Loading required package: Biostrings

## Loading required package: XVector

library(BSgenome.Mmusculus.UCSC.mm9)  
library(Biostrings)  
library(tibble)  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Biostrings':  
##   
## collapse, intersect, setdiff, setequal, union

## The following object is masked from 'package:XVector':  
##   
## slice

## The following objects are masked from 'package:GenomicRanges':  
##   
## intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':  
##   
## intersect

## The following objects are masked from 'package:IRanges':  
##   
## collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':  
##   
## first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':  
##   
## combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(reshape2)  
library("Homo.sapiens")

## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

##   
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':  
##   
## select

## Loading required package: OrganismDbi

## Loading required package: GenomicFeatures

## No methods found in "RSQLite" for requests: dbGetQuery

## Loading required package: GO.db

##

## Loading required package: org.Hs.eg.db

##

## Loading required package: TxDb.Hsapiens.UCSC.hg19.knownGene

library(pander)  
library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':  
##   
## smiths

## The following object is masked from 'package:magrittr':  
##   
## extract

## The following object is masked from 'package:S4Vectors':  
##   
## expand

library(GenomicInteractions)

## Loading required package: InteractionSet

## Loading required package: SummarizedExperiment

library(AnnotationHub)

##   
## Attaching package: 'AnnotationHub'

## The following object is masked from 'package:Biobase':  
##   
## cache

library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:GenomicInteractions':  
##   
## is.trans

library(Biobase)  
library(readxl)  
library(readr)

##   
## Attaching package: 'readr'

## The following object is masked from 'package:scales':  
##   
## col\_factor

enhancers<-import("~/DataFiles/Enhancer Tracks/Human/human\_permissive\_enhancers\_phase\_1\_and\_2.bed")  
genes<-import("~/DataFiles/Gene Tracks/Human/hg.bed")  
promotersGenes<-promoters(genes)  
  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
ARXTandem2SpacedTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARXTande2SpacedSites")  
Plaindromic4SpacedTFBS<-readRDS("~/DataFiles/ChIPseq/Human/Plaindromic4SpacedTFBS")  
JolmaTFBS<-readRDS("~/DataFiles/ChIPseq/Human/JolmaTFBS")  
  
ARXMotifModelList<-c("6 Mer"= arx6merTFBS,  
 "Tandem 2 Spaced" = ARXTandem2SpacedTFBS,  
 "Palindromic 4 Spaced" = Plaindromic4SpacedTFBS,  
 "Jolma Model" = JolmaTFBS)  
  
arxPromoters<-lapply(ARXMotifModelList, function(x)c(subsetByOverlaps(x, promotersGenes))%>%unlist)  
arxEnhancers<-lapply(ARXMotifModelList, function(x)c(subsetByOverlaps(x, enhancers))%>%unlist)  
phyloPScores<-lapply(arxPromoters, function(x){import("~/DataFiles/Conservation/Human/hg19.100way.phyloP100way.bw", which= x)})  
phyloPAbove0.5<-lapply(phyloPScores, function(x){subset(x, score>=0.5)})  
conservedARXPromoters<-list()  
for(i in 1:4){  
 conservedARXPromoters[i]<-subset(arxPromoters[[i]], countOverlaps(arxPromoters[[i]], phyloPAbove0.5[[i]])>=6)  
}  
  
names(conservedARXPromoters)<-names(ARXMotifModelList)  
conservedARXPromotersAndEnhancers<-list(c(conservedARXPromoters$`6 Mer`, arxEnhancers$`6 Mer`)%>%unlist(),  
 c(conservedARXPromoters$`Tandem 2 Spaced`, arxEnhancers$`Tandem 2 Spaced`)%>%unlist(),  
 c(conservedARXPromoters$`Palindromic 4 Spaced`, arxEnhancers$`Palindromic 4 Spaced`)%>%unlist(),  
 c(conservedARXPromoters$`Jolma Model`, arxEnhancers$`Jolma Model`)%>%unlist()  
 )  
  
names(conservedARXPromotersAndEnhancers)<-names(ARXMotifModelList)

## Bar graph Human

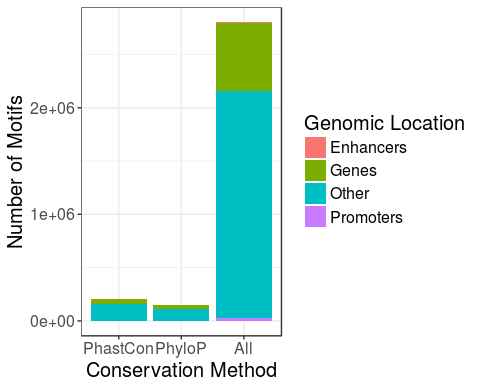
fantom5Enhancers<- import("~/DataFiles/Enhancer Tracks/Human/human\_permissive\_enhancers\_phase\_1\_and\_2.bed")  
hg19Genes<-import("~/DataFiles/Gene Tracks/Human/hg.bed")  
hg19Promoters<-promoters(hg19Genes)  
  
# genome<-BSgenome.Hsapiens.UCSC.hg19  
# arx6merTFBS<-matchPWM(round(PWM("TAATTA")\*7), genome, "100%")  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
  
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>=0.5)  
  
##See which Motifs fall into these regions  
polyPConserved<-subset(arx6merTFBS,countOverlaps( arx6merTFBS,phyloPTrack)>=6)  
  
##Phast Con Scores Same as above  
phastConScores<-import("~/DataFiles/Conservation/Human/hg19.100way.phastCons.bw",which =arx6merTFBS)  
  
PhastConTrack<-subset(phastConScores, score>=0.85)  
  
##See which Motifs fall into these regions  
phastConserved<-subset(arx6merTFBS,countOverlaps(arx6merTFBS, PhastConTrack)>=6)  
  
  
  
genomicLocation<-function(x){  
 dataFrame1<-rbind.data.frame(  
 "Enhancers" =subsetByOverlaps(x, fantom5Enhancers)%>%length(),  
   
 "Genes" = subsetByOverlaps(x, hg19Genes)%>%length(),  
   
 "Promoters"= subsetByOverlaps(x, hg19Promoters)%>%length(),  
   
 "Other"= length(x)-sum(  
 "Enhancers" =subsetByOverlaps(x, fantom5Enhancers)%>%length(),  
 "Genes" = subsetByOverlaps(x, hg19Genes)%>%length(),  
 "Promoters"= subsetByOverlaps(x, hg19Promoters)%>%length()))  
   
 colnames(dataFrame1) <-c("Number Of Motifs")  
 return(dataFrame1)  
}  
  
ARXMotifs<-cbind.data.frame(genomicLocation(phastConserved),   
 genomicLocation(polyPConserved),  
 genomicLocation(arx6merTFBS))%>%rownames\_to\_column()  
colnames(ARXMotifs)<-c("Genomic Location",  
 "PhastCon",  
 "PhyloP",   
 "All")  
  
ARXMotifs%>%pander()

|  |  |  |  |
| --- | --- | --- | --- |
| Genomic Location | PhastCon | PhyloP | All |
| Enhancers | 1838 | 1372 | 10544 |
| Genes | 50391 | 35334 | 638109 |
| Promoters | 2470 | 1775 | 28642 |
| Other | 153177 | 110297 | 2125071 |

write.table(x = ARXMotifs,file = "~/Thesis/ConservationMotifs",quote = FALSE, append= FALSE, sep = "\t")  
  
reshapedConservationScoresAtlocations<-ARXMotifs%>%melt()

## Using Genomic Location as id variables

ggplot(data = reshapedConservationScoresAtlocations, mapping = aes(x=`variable`, y= `value`, fill= `Genomic Location`))+  
 geom\_bar(stat="identity")+  
 theme\_bw()+  
 ylab(label = "Number of Motifs")+  
 xlab(label = "Conservation Method")+  
 guides(fill=guide\_legend(title= "Genomic Location"))+  
 theme(text = element\_text(size = 15))



## Heat Map of Conservation

#Data Imports  
enhancers<-import("~/DataFiles/Enhancer Tracks/Human/human\_permissive\_enhancers\_phase\_1\_and\_2.bed")  
genes<-import("~/DataFiles/Gene Tracks/Human/hg.bed")  
promotersGenes<-promoters(genes)  
  
arx6merTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARX6merHg19Sites")  
ARXTandem2SpacedTFBS<-readRDS("~/DataFiles/ChIPseq/Human/ARXTande2SpacedSites")  
Plaindromic4SpacedTFBS<-readRDS("~/DataFiles/ChIPseq/Human/Plaindromic4SpacedTFBS")  
JolmaTFBS<-readRDS("~/DataFiles/ChIPseq/Human/JolmaTFBS")  
  
ARXMotifModelList<-c("6 Mer"= arx6merTFBS,  
 "Tandem 2 Spaced" = ARXTandem2SpacedTFBS,  
 "Palindromic 4 Spaced" = Plaindromic4SpacedTFBS,  
 "Jolma Model" = JolmaTFBS)  
  
  
  
#PhyloP imports  
phyloPScores<-lapply(ARXMotifModelList, function(x){import("~/DataFiles/Conservation/Human/hg19.100way.phyloP100way.bw",  
 which= x)})

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

phyloPAbove0.5<-lapply(phyloPScores, function(x){subset(x, score>=0.5)})  
  
  
#Conserved phyloP  
conservedARXPromotersAndEnhancers<-list()  
for(i in 1:4){  
 conservedARXPromotersAndEnhancers[i]<-subset(ARXMotifModelList[[i]], countOverlaps(ARXMotifModelList[[i]], phyloPAbove0.5[[i]])>=6)  
}

## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated

## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated  
  
## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated  
  
## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated

## PhastCon Import scores  
PhastConScores<-lapply(ARXMotifModelList, function(x){import("~/DataFiles/Conservation/Human/hg19.100way.phastCons.bw",  
 which= x)})

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

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

PhastConAbove0.85<-lapply(phyloPScores, function(x){subset(x, score>=0.85)})  
  
#PhastCon Conserved Motifs  
conservedARXPromotersAndEnhancersPhastCon<-list()  
for(i in 1:4){  
 conservedARXPromotersAndEnhancersPhastCon[i]<-subset(ARXMotifModelList[[i]],  
 countOverlaps(ARXMotifModelList[[i]], PhastConAbove0.85[[i]])>=6)  
}

## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated

## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated  
  
## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated  
  
## Warning in `[<-`(`\*tmp\*`, i, value = <S4 object of class  
## structure("GRanges", package = "GenomicRanges")>): implicit list embedding  
## of S4 objects is deprecated

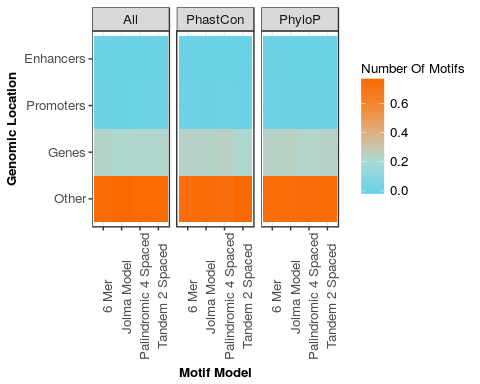
## Proportions of motifs in various genomic Regions  
MotifModelGenomicLocation<-lapply(ARXMotifModelList, function(x){  
 dataFrameGenomicLocation<-rbind.data.frame("Enhancers"=subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 "Genes"=subsetByOverlaps(x, genes)%>%length() /length(x) ,  
 "Promoters"=subsetByOverlaps(x, promotersGenes)%>%length()/length(x),  
 "Other"=(length(x)-sum(subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 subsetByOverlaps(x, genes)%>%length(),  
 subsetByOverlaps(x, promotersGenes)%>%length()))/length(x))  
 set\_colnames(dataFrameGenomicLocation,value = "Number Of Motifs")  
})  
  
  
  
PhyloPConservedGenomicLocation<-lapply(conservedARXPromotersAndEnhancers, function(x){  
 dataFrameGenomicLocation<-rbind.data.frame("Enhancers"=subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 "Genes"=subsetByOverlaps(x, genes)%>%length() /length(x) ,  
 "Promoters"=subsetByOverlaps(x, promotersGenes)%>%length()/length(x),  
 "Other"=(length(x)-sum(subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 subsetByOverlaps(x, genes)%>%length(),  
 subsetByOverlaps(x, promotersGenes)%>%length()))/length(x))  
 set\_colnames(dataFrameGenomicLocation,value = "Number Of Motifs")  
})  
  
  
  
  
  
PhastConservedGenomicLocation<-lapply(conservedARXPromotersAndEnhancersPhastCon, function(x){  
 dataFrameGenomicLocation<-rbind.data.frame("Enhancers"=subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 "Genes"=subsetByOverlaps(x, genes)%>%length() /length(x) ,  
 "Promoters"=subsetByOverlaps(x, promotersGenes)%>%length()/length(x),  
 "Other"=(length(x)-sum(subsetByOverlaps(x, enhancers)%>%length()/length(x),  
 subsetByOverlaps(x, genes)%>%length(),  
 subsetByOverlaps(x, promotersGenes)%>%length()))/length(x))  
 set\_colnames(dataFrameGenomicLocation,value = "Number Of Motifs")  
})  
  
  
#Naming them again as they're in order and we can/should  
names(PhyloPConservedGenomicLocation)<-names(ARXMotifModelList)  
names(PhastConservedGenomicLocation)<-names(ARXMotifModelList)  
names(MotifModelGenomicLocation)<-names(ARXMotifModelList)  
  
  
#One dataframe one love  
AllMotifModels<-do.call(rbind.data.frame,c("PhyloP"=PhyloPConservedGenomicLocation,  
 "PhastCon"=PhastConservedGenomicLocation,  
 "All"=MotifModelGenomicLocation))%>%rownames\_to\_column(var= "Names")  
  
#Spilliting the first column with all the informaiton so we can facet it  
AllMotifModels<-separate(data = AllMotifModels,col = "Names",into = c("Conservation Method","Motif Model", "Genomic Location"), sep = '\\.' )  
  
## Because theres numbers sometimes we need to remove  
AllMotifModels<-separate(data = AllMotifModels,col = "Genomic Location",into = c("Genomic Location", "Numbers" ),sep = "1")

## Warning: Too few values at 48 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,  
## 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

AllMotifModels<-separate(data = AllMotifModels,col = "Genomic Location",into = c("Genomic Location", "Numbers" ),sep = "2")

## Warning: Too few values at 48 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,  
## 12, 13, 14, 15, 16, 17, 18, 19, 20, ...

#ORdering le plot  
AllMotifModels$`Genomic Location`<-factor(AllMotifModels$`Genomic Location`, levels = c("Other", "Genes", "Promoters", "Enhancers"))  
  
  
## Plotting  
ggplot(AllMotifModels)+  
 geom\_tile(aes(x=`Motif Model`, y= `Genomic Location`, fill= `Number Of Motifs` ))+  
 facet\_wrap(~`Conservation Method`)+  
 xlab(label = "Motif Model")+  
 ylab(label = "Genomic Location")+  
 theme\_bw()+  
 theme(axis.text.x = element\_text(size= 10,angle = 90),  
 axis.text.y= element\_text(size= 10),  
 axis.title = element\_text(size=10,face = "bold"),  
 legend.text = element\_text(size=10),   
 legend.title = element\_text(size=10),   
 strip.text = element\_text(size = 10))+  
 scale\_fill\_gradientn(colours=c("#69D2E7", "#A7DBD8",   
 "#F38630", "#FA6900"),  
 values=rescale(c( -2, -1,  
 1, 2)),  
 guide="colorbar")



fetalBrainMaleHMM<-import("~/DataFiles/ChromHMM/human/E081\_15\_coreMarks\_dense.bed.gz")  
  
  
  
  
conservedARXPromotersAndEnhancersGrange<-c(conservedARXPromotersAndEnhancers[[1]],  
 conservedARXPromotersAndEnhancers[[2]],  
 conservedARXPromotersAndEnhancers[[3]],  
 conservedARXPromotersAndEnhancers[[4]])  
activeMotifs<-subset(fetalBrainMaleHMM, name %in% c( "5\_TxWk",   
 "4\_Tx",  
 "1\_TssA",   
 "3\_TxFlnk",  
 "2\_TssAFlnk",   
 "7\_Enh",  
 "6\_EnhG",  
 "12\_EnhBiv",   
 "11\_BivFlnk",   
 "10\_TssBiv" ))%>%subsetByOverlaps(conservedARXPromotersAndEnhancersGrange, .)  
  
PromotersWithAccessibleARXMotifs<-subsetByOverlaps(promotersGenes,activeMotifs)  
  
UCSCConvter <- read\_delim("~/DataFiles/Gene Tracks/Human/hg19WithNames.bed",   
 "\t", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## hg19.knownGene.txStart = col\_integer(),  
## hg19.knownGene.txEnd = col\_integer(),  
## hg19.knownGene.cdsStart = col\_integer(),  
## hg19.knownGene.cdsEnd = col\_integer(),  
## hg19.knownGene.exonCount = col\_integer(),  
## hg19.knownGene.exonStarts = col\_number(),  
## hg19.knownGene.exonEnds = col\_number()  
## )

## See spec(...) for full column specifications.

#We can left join based on the transcript Ids from the promoter id's gene  
GeneSymbolTHesisTable<-left\_join(cbind.data.frame("Promoter Ids"=PromotersWithAccessibleARXMotifs$name), UCSCConvter,   
 by= c("Promoter Ids" = "hg19.kgXref.kgID" ))

## Warning: Column `Promoter Ids`/`hg19.kgXref.kgID` joining factor and  
## character vector, coercing into character vector

GenesWithARXMotifs<-cbind.data.frame(GeneSymbolTHesisTable$hg19.kgXref.geneSymbol,  
 GeneSymbolTHesisTable$hg19.knownGene.chrom,  
 GeneSymbolTHesisTable$hg19.knownGene.txStart,  
 GeneSymbolTHesisTable$hg19.knownGene.txEnd,  
 GeneSymbolTHesisTable$`Promoter Ids`)  
  
UniqueGenesWithARXMotifs<-GenesWithARXMotifs[isUnique(GenesWithARXMotifs$`GeneSymbolTHesisTable$hg19.kgXref.geneSymbol`),]  
  
colnames(UniqueGenesWithARXMotifs)<-c("GeneSymbol", "chr", "start", "end", "transcriptId")

### Significant Human interactions

## Read in data tables  
GiTrack<-readRDS("~/DataFiles/HiC/Human/StasticallySignificantHg19BetweenPromoterEnhancer")  
dataFrameGiTrack<-GiTrack%>%as.data.frame()  
  
  
## Split Them up based on anchor location eg Anchor 1 is promoter anchor 2 is Enhancer or enhancer is anchor 1 and promoter is anchor 2.   
Spilt1<-subset(dataFrameGiTrack, !promoters.id1=="NA")  
Spilt2<-subset(dataFrameGiTrack, !promoters.id2=="NA")  
  
  
Spilt1<-cbind.data.frame("PromoterIds"=Spilt1$promoters.id1,  
 "EnhancerIds"= Spilt1$enhancers.id2,  
 "Counts" = Spilt1$counts,   
 "q\_value" = Spilt1$SignificantValues.q\_value)   
  
Split2<-cbind.data.frame("PromoterIds"=Spilt2$promoters.id2,  
 "EnhancerIds"= Spilt2$enhancers.id1,  
 "Counts" = Spilt2$counts,   
 "q\_value" = Spilt2$SignificantValues.q\_value)   
  
  
#Combine the two Datafmraes into 1  
PromoterIds<-rbind(Spilt1,Split2)  
  
#Convert from ASIS format to dataframe  
PromoterIds$PromoterIds<-lapply(PromoterIds$PromoterIds, list)  
PromoterIds$PromoterIds<-lapply(PromoterIds$PromoterIds,function(x){do.call(rbind, x)})  
  
  
  
  
##Paste Enhancers for each promoter   
dataFrame2<-NULL  
for(i in 1:dim(PromoterIds)[1]){  
 test<-PromoterIds[i,]  
   
 dataFrame1<-NULL  
 for(t in 1:(test$PromoterIds%>%as.data.frame%>%dim)[2]){  
 dataFrame1<-rbind(dataFrame1,test[2:4])  
 }  
   
 dataFrame2<-rbind.data.frame(dataFrame2,  
 cbind.data.frame("Promoter Ids" =t(test$PromoterIds%>%as.data.frame()),  
 dataFrame1)  
 )  
}  
dataFrame2$`Promoter Ids`<-dataFrame2$`Promoter Ids`%>%as.character()  
  
  
  
  
  
#Convert from UCSC to gene symbol because biomart wasn't working/ was only returning maybe 5-10 genes out of the 50!  
  
##Bedfile From ucsc with all the metadata columns ;-)   
UCSCConvter <- read\_delim("~/DataFiles/Gene Tracks/Human/hg19WithNames.bed",   
 "\t", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## hg19.knownGene.txStart = col\_integer(),  
## hg19.knownGene.txEnd = col\_integer(),  
## hg19.knownGene.cdsStart = col\_integer(),  
## hg19.knownGene.cdsEnd = col\_integer(),  
## hg19.knownGene.exonCount = col\_integer(),  
## hg19.knownGene.exonStarts = col\_number(),  
## hg19.knownGene.exonEnds = col\_number()  
## )

## See spec(...) for full column specifications.

#We can left join based on the transcript Ids from the promoter id's gene  
GeneSymbolTHesisTable<-left\_join(dataFrame2, UCSCConvter, by= c("Promoter Ids" = "hg19.kgXref.kgID" ))  
  
  
#Now we can make a table  
EnhancerPromoterInteractionScoreCounts<-cbind.data.frame("Gene Symbol"=GeneSymbolTHesisTable$hg19.kgXref.geneSymbol,  
 "UCSC Transcript Id"= GeneSymbolTHesisTable$`Promoter Ids`,  
 "Chromosome"= GeneSymbolTHesisTable$hg19.knownGene.chrom,  
 "Start" = GeneSymbolTHesisTable$hg19.knownGene.txStart,  
 "End" =GeneSymbolTHesisTable$hg19.knownGene.txEnd,  
 "Enhancer"=GeneSymbolTHesisTable$EnhancerIds,   
 "Q\_value"=GeneSymbolTHesisTable$q\_value)  
  
#Subset for unique Ids  
UniqueInteractionsBetweenHumansAndGeneSYmbo<-EnhancerPromoterInteractionScoreCounts[isUnique(EnhancerPromoterInteractionScoreCounts$`Gene Symbol`),]  
#TABLE  
  
write.table(UniqueInteractionsBetweenHumansAndGeneSYmbo, file = "~/Thesis/uniquePromoterEnhancersHuman",  
 append=FALSE,  
 quote=FALSE,   
 col.names=TRUE,  
 row.names = FALSE,  
 sep = "\t")

## Identifying how many genes in humans have cosnerved ARX TFBS: 3. We got 3 out of the 28 genes

RNASeqHumanGenes<-read\_delim("~/DataFiles/RNAseq/Human/orthologuesLiftOverPA1",   
 "\t", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## `Output Row Number` = col\_integer(),  
## `Input Order` = col\_integer(),  
## `% match of input gene` = col\_integer(),  
## `% match of target (reciprocal match)` = col\_integer(),  
## `Input Start Coord` = col\_integer(),  
## `Input End Coord` = col\_integer(),  
## `Input Strand` = col\_integer(),  
## `Input Entrez ID` = col\_integer(),  
## `Target Start Coord` = col\_integer(),  
## `Target End Coord` = col\_integer(),  
## `Target Strand` = col\_integer(),  
## `Target Entrez ID` = col\_integer()  
## )

## See spec(...) for full column specifications.

#RNASeqHumanGenes[isUnique(RNASeqHumanGenes$HGNC.symbol),]  
genesWithARXMotifs<-left\_join(RNASeqHumanGenes, UniqueGenesWithARXMotifs, by = c("Target Common Name"= "GeneSymbol"))

## Warning: Column `Target Common Name`/`GeneSymbol` joining character vector  
## and factor, coercing into character vector

DifferentiallyExpressedGenes<-genesWithARXMotifs[!is.na(genesWithARXMotifs$end),]  
  
GenesDifferentiallyExpressedUnique<-DifferentiallyExpressedGenes[isUnique(DifferentiallyExpressedGenes$HGNC.symbol),]

## Warning: Unknown or uninitialised column: 'HGNC.symbol'.

Only getting 3-10 genes being differentialy expressed. Hence indicating this is probably not the best mechanism.

# Number of Genes interacting with enhancer motifs

library(readr)  
EnhancerPromoterInteractionsHuman <- read\_delim("~/Thesis/EnhancerPromoterInteractionsHuman",   
 " ", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## Gene = col\_character(),  
## Symbol = col\_character(),  
## UCSC = col\_character(),  
## Transcript = col\_character(),  
## Id = col\_character(),  
## Chromosome = col\_character(),  
## Start = col\_character(),  
## End = col\_character(),  
## Enhancer = col\_double(),  
## Q\_value = col\_character()  
## )

## Warning in rbind(names(probs), probs\_f): number of columns of result is not  
## a multiple of vector length (arg 1)

## Warning: 289 parsing failures.  
## row # A tibble: 5 x 5 col row col expected actual expected <int> <chr> <chr> <chr> actual 1 1 <NA> 10 columns 7 columns file 2 2 <NA> 10 columns 7 columns row 3 3 <NA> 10 columns 7 columns col 4 4 <NA> 10 columns 7 columns expected 5 5 <NA> 10 columns 7 columns actual # ... with 1 more variables: file <chr>  
## ... ................. ... .................................. ........ .................................. ...... .................................. .... .................................. ... .................................. ... .................................. ........ .................................. ...... .......................................  
## See problems(...) for more details.

genesWithARXMotifs<-left\_join(RNASeqHumanGenes, EnhancerPromoterInteractionsHuman, by = c("Target Common Name"= "Gene"))  
  
DifferentiallyExpressedGenes<-genesWithARXMotifs[!is.na(genesWithARXMotifs$end),]

## Warning: Unknown or uninitialised column: 'end'.

## Warning in is.na(genesWithARXMotifs$end): is.na() applied to non-(list or  
## vector) of type 'NULL'

ANd 0 Overlap with genes in humans.

Not sure if this shows lack of conservation of targets or shows lack of conservation of binding and therefore subject of productive. A big contributot is that the mechanisms i used to lift over the gene lists 1; bioMart rertunred only 40 genes out of the 825.

And the other mechanism which was using Online converter returned 400 but again less than half of the total number.