Limma ChIP-chip Calling Script

Awais Choudhry

07/09/2017

library(limma)  
library(Ringo)

## Loading required package: Biobase

## 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 object is masked from 'package:limma':  
##   
## plotMA

## 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

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

## Loading required package: RColorBrewer

## Loading required package: Matrix

## Loading required package: grid

## Loading required package: lattice

library(edgeR)  
library(magrittr)  
library(tidyr)

##   
## Attaching package: 'tidyr'

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

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

library(rtracklayer)

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: S4Vectors

##   
## Attaching package: 'S4Vectors'

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

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

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

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

##   
## Attaching package: 'GenomeInfoDb'

## The following objects are masked from 'package:Ringo':  
##   
## genome, genome<-

library(BSgenome.Mmusculus.UCSC.mm9)

## Loading required package: BSgenome

## Loading required package: Biostrings

## Loading required package: XVector

library(Biostrings)  
library(pander)

arrayfiles <- list.files(path="/home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent016/N2aCells/",  
 pattern="txt.gz")  
RG <- read.maimages(arrayfiles,  
 source="agilent",  
 path="/home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent016/N2aCells/")

## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent016/N2aCells//GSM742100\_US45103054\_251471611506\_S01\_ChIP.txt.gz   
## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent016/N2aCells//GSM742101\_US45103054\_251471611507\_S01\_ChIP.txt.gz   
## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent016/N2aCells//GSM742102\_US45103054\_251471611508\_S01\_ChIP.txt.gz

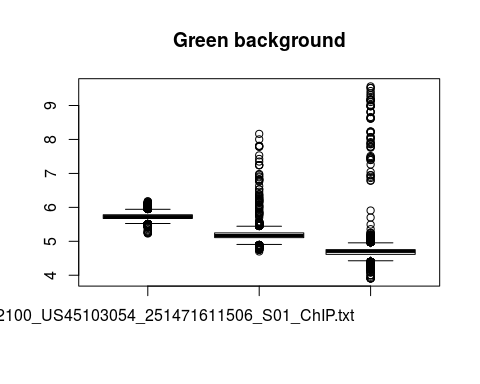
I have spilt the ChIP-chip data into 2 seperate files based on the day they were processed. I believe is this is due to the datasets being ran on different platforms. The impilications of this i am unaware of.

## Annotating the files

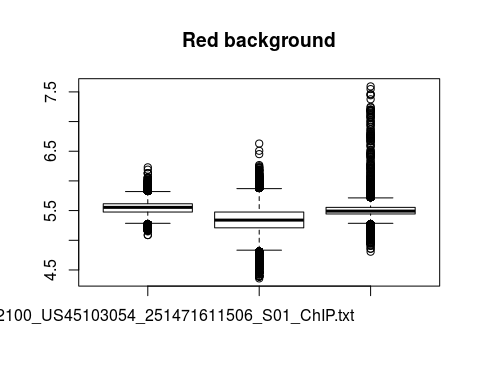
###################################################  
### code chunk number 27: readAgiTargets  
###################################################  
at <- readTargets(file.path("/home/a1649239/DataFiles/ChIPseq/Mouse/","filelist.txt"))[1:3,] # the subset at the end is my edit  
RG$targets <- at

# BackGround Intensities

boxplot(data.frame(log2(RG$Gb)),main="Green background")



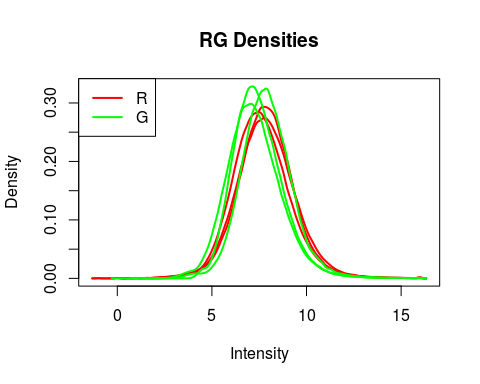
boxplot(data.frame(log2(RG$Rb)),main="Red background")



## Preprocessing

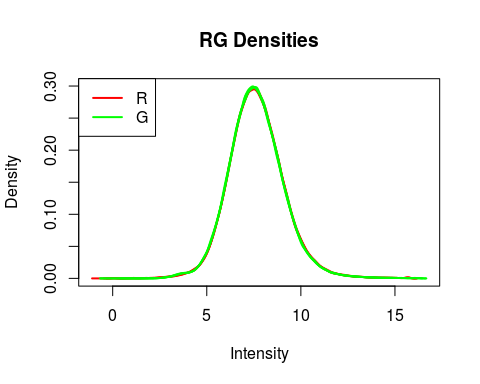
MA <- normalizeWithinArrays(RG, method="loess")  
  
plotDensities(RG)

## Warning in plotDensities.RGList(RG): NaNs produced

 Consequence of this plot we need to correct for background

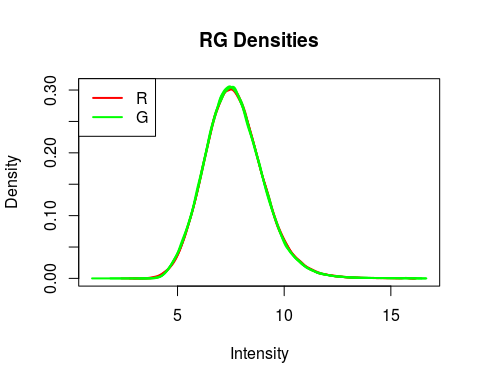
## Normalizing between Arrays

MAq <- normalizeBetweenArrays(MA, method = "Aquantile")  
 plotDensities(MAq)



## Removing Negatively/down regulated probes

MAqNegativeControlRemoved<-MAq[MAq$genes$ControlType=="0",]  
  
  
LowlyExpressedAndNegativeControlsRemoved<-MAqNegativeControlRemoved[rowSums(MAqNegativeControlRemoved$A>=4.2)>=3,]  
  
normalizedLowlyExpressedAndNegativeControlsRemoved<-normalizeBetweenArrays(LowlyExpressedAndNegativeControlsRemoved, method="Aquantile")  
  
plotDensities(normalizedLowlyExpressedAndNegativeControlsRemoved)



fit <- lmFit(normalizedLowlyExpressedAndNegativeControlsRemoved)  
fit <- eBayes(fit)  
summary(decideTests(fit))

## x1  
## -1 5697  
## 0 223164  
## 1 7098

ARXEnrichedN2a1<-topTable(fit, number = summary(decideTests(fit))[3])

##### ARRAY 2 Of the Whole Brain.

###################################################  
### code chunk number 26: readAgilentData  
###################################################  
arrayfiles <- list.files(path="/home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent017/N2a/",  
 pattern="txt.gz")  
RG <- read.maimages(arrayfiles,  
 source="agilent",  
 path="/home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent017/N2a/")

## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent017/N2a//GSM742106\_US45103054\_251471711563\_S01\_ChIP.txt.gz   
## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent017/N2a//GSM742107\_US45103054\_251471711562\_S01\_ChIP.txt.gz   
## Read /home/a1649239/DataFiles/ChIPseq/Mouse/ChIPAngilent017/N2a//GSM742108\_US45103054\_251471711561\_S01\_ChIP.txt.gz

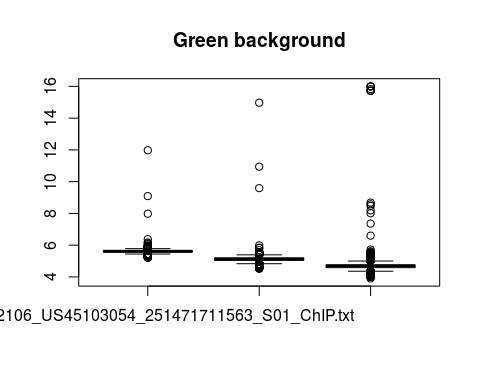
Because the ChIP-chip arrays are spilt based on different chromosomes I am calling the peaks from the same array and sample.

## Annotating the files

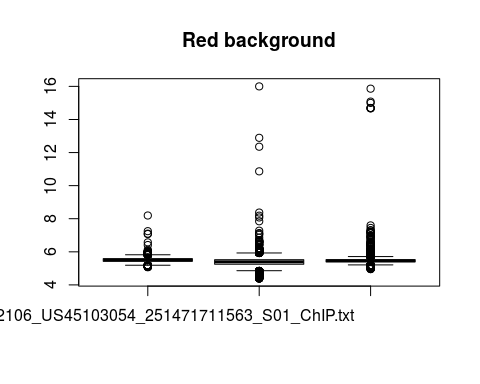
###################################################  
### code chunk number 27: readAgiTargets  
###################################################  
at <- readTargets(file.path("/home/a1649239/DataFiles/ChIPseq/Mouse/","filelist.txt"))[7:9,] # the subset at the end is my edit  
RG$targets <- at

# BackGround Intensities

boxplot(data.frame(log2(RG$Gb)),main="Green background")



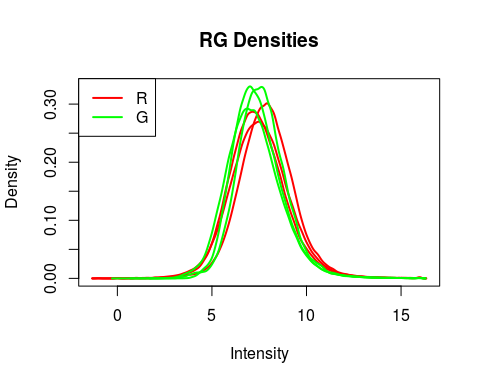
boxplot(data.frame(log2(RG$Rb)),main="Red background")



## Preprocessing

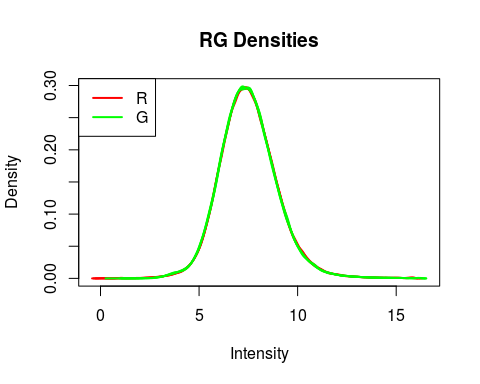
MA <- normalizeWithinArrays(RG, method="loess")  
  
plotDensities(RG)

## Warning in plotDensities.RGList(RG): NaNs produced

 Consequence of this plot we need to correct for background

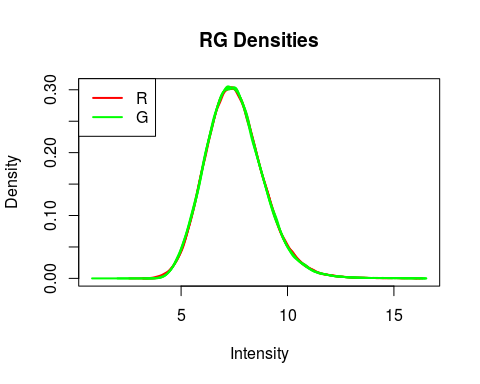
## Normalizing between Arrays

MAq <- normalizeBetweenArrays(MA, method = "Aquantile")  
 plotDensities(MAq)



## Removing Negatively/down regulated probes

MAqNegativeControlRemoved<-MAq[MAq$genes$ControlType=="0",]  
  
  
LowlyExpressedAndNegativeControlsRemoved<-MAqNegativeControlRemoved[rowSums(MAqNegativeControlRemoved$A>=4.2)>=3,]  
  
normalizedLowlyExpressedAndNegativeControlsRemoved<-normalizeBetweenArrays(LowlyExpressedAndNegativeControlsRemoved, method="Aquantile")  
  
plotDensities(normalizedLowlyExpressedAndNegativeControlsRemoved)



# PositiveProbesOnly<-normalizedLowlyExpressedAndNegativeControlsRemoved[!normalizedLowlyExpressedAndNegativeControlsRemoved$genes$ProbeName %in% NegativeProbes$ProbeName,]  
#   
# plotDensities(PositiveProbesOnly)

fit <- lmFit(normalizedLowlyExpressedAndNegativeControlsRemoved)  
fit <- eBayes(fit)  
summary(decideTests(fit))

## x1  
## -1 2806  
## 0 229275  
## 1 3496

ARXEnrichedN2a2<-topTable(fit, number = "inf")

## Combing The ChIP enriched regions and removing negative controls and selecting for positively expressing probes

N2aEnrichedChIPRegions<-rbind.data.frame(ARXEnrichedN2a1  
 ,ARXEnrichedN2a2)  
  
N2aEnrichedChIPRegionsSignificant<-subset(N2aEnrichedChIPRegions, adj.P.Val<=0.001)  
  
PositiveProbes<-subset(N2aEnrichedChIPRegionsSignificant, logFC>=0)  
  
PositiveProbes<-subset(PositiveProbes, !SystematicName== "unmapped")

# Identifying how many of peaks contain an Arx motif

ChiPN2aGRANGEs<-separate(PositiveProbes,col = "SystematicName", into= c("chromosome", "start", "end") )  
  
ChiPN2aGRANGEs<-subset(ChiPN2aGRANGEs, start<end)  
  
  
## Making them 400bp+ Regions centered on the probe  
EnrichedChIPRegions<-(makeGRangesFromDataFrame(ChiPN2aGRANGEs, keep.extra.columns = TRUE)+200)  
  
  
  
### Intersecting with motif sites  
JolmaSites<-matchPWM(pwm = cbind(round(PWM("TAATT")\*4), 0.25, round(PWM("ATTAA")\*4)), BSgenome.Mmusculus.UCSC.mm9, "80%")

## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them

## Warning in .Call2("PWM\_score\_starting\_at", pwm, subject, starting.at,  
## base\_codes, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them

## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them  
  
## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them

## Warning in .Call2("PWM\_score\_starting\_at", pwm, subject, starting.at,  
## base\_codes, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them

## Warning in .Call2("XString\_match\_PWM", pwm, subject, min.score,  
## count.only, : 'subject' contains letters not in [ACGT] ==> assigned weight  
## 0 to them

arx4merSites<-readRDS("~/DataFiles/ChIPseq/Mouse/ARX4mermm9Sites")  
arxSites<-readRDS("~/DataFiles/ChIPseq/Mouse/ARX6mermm9Sites")  
  
rbind.data.frame(  
"6mers In peaks"=subsetByOverlaps(EnrichedChIPRegions, arxSites)%>%length(),  
"Jolma In Peaks"= subsetByOverlaps(EnrichedChIPRegions, JolmaSites)%>%length(),  
"4mers In Peaks" = subsetByOverlaps(EnrichedChIPRegions, arx4merSites)%>%length(),  
"Total number of Peaks"= EnrichedChIPRegions%>%length()  
)%>%set\_colnames("Number Of Peaks With Motif")%>%pander()

|  |  |
| --- | --- |
|  | Number Of Peaks With Motif |
| **6mers In peaks** | 44 |
| **Jolma In Peaks** | 109 |
| **4mers In Peaks** | 196 |
| **Total number of Peaks** | 219 |