Script for over-representation

Awais Choudhry

09/10/2017

## Checking to see if the numbers are robust  
  
library(magrittr)  
library(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(ggplot2)  
library(magrittr)  
library(tibble)  
library(pander)  
library(reshape2)  
library(plyr)

##   
## Attaching package: 'plyr'

## The following object is masked from 'package:IRanges':  
##   
## desc

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

library(MotifDb)

## Loading required package: Biostrings

## Loading required package: XVector

##   
## Attaching package: 'XVector'

## The following object is masked from 'package:plyr':  
##   
## compact

## See system.file("LICENSE", package="MotifDb") for use restrictions.

library(magrittr)  
library(reshape2)  
library(BSgenome.Hsapiens.UCSC.hg19)

## Loading required package: BSgenome

## Loading required package: rtracklayer

library(BSgenome.Mmusculus.UCSC.mm9)

# Mouse Inputs

###Inputs!  
enhancerGrange <-  
 import(con = "~/DataFiles/Enhancer Tracks/Mouse/mouse\_permissive\_enhancers\_phase\_1\_and\_2.bed")  
UCSCgenes <- import("~/DataFiles/Gene Tracks/Mouse/mm9.bed")  
startSites<-subset(import("~/DataFiles/Gene Tracks/Mouse/FullMm9genome.GTF"), type== "start\_codon")  
promoters <- promoters(UCSCgenes)  
genome<-BSgenome.Mmusculus.UCSC.mm9

# Human Inputs

###Inputs!  
enhancerGrange <-  
 import(con = "~/DataFiles/Enhancer Tracks/Human/human\_permissive\_enhancers\_phase\_1\_and\_2.bed")  
UCSCgenes <- import("~/DataFiles/Gene Tracks/Human/hg.bed")  
startSites<-subset(import("~/DataFiles/Gene Tracks/Human/hg19.gtf"), type== "start\_codon")  
promoters <- promoters(UCSCgenes)  
genome<-BSgenome.Hsapiens.UCSC.hg19

### The script to make the plots

To do this, the script will 1. Generate the PWM 2. Match the genome at a 100% match rate 3. Overlap with annotated enhancers, promoters, genes, and then other regions and count the number for each region 4. Generate a data table and plot the table.

With the original Granges we also calculated the distance to the closest TSS which was defined as the start codon. and then merged each of these into 1 data table that we plotted.

## Generate the PWMs  
ArxPlaindrmicMinus1<-rbind( A=c(0,1,1,0,0,0,0,1,1,0),   
 C=c(0,0,0,0,0,0,0),  
 G=c(0,0,0,0,0,0,0),  
 T=c(1,0,0,1,1,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWMNospace<-rbind( A=c(0,1,1,0,0,1,0,1,1,0,0,1),   
 C=c(0,0,0,0,0,0,0),  
 G=c(0,0,0,0,0,0,0) ,  
 T=c(1,0,0,1,1,0,1,0,0,1,1,0))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM1space<-rbind( A=c(0,1,1,0,0,1,0.25,1,0,0,1,1,0),   
 C=c(0,0,0,0,0,0,0.25,0),  
 G=c(0,0,0,0,0,0,0.25,0),  
 T=c(1,0,0,1,1,0,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM2space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,1,0,0,1,1,0),   
 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(1,0,0,1,1,0,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM3space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,1,0,0,1,1,0),  
 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(1,0,0,1,1,0,0.25,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM4space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,1,0,0,1,1,0),  
 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(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM5space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0),  
 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(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM6space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0),  
 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(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

arx6MerPWM7space<-rbind( A=c(0,1,1,0,0,1,0.25,0.25,0.25,0.25,0.25,0.25,0.25,1,0,0,1,1,0),  
 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(1,0,0,1,1,0,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0,1,1,0,0,1))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

### Tandeom Sites  
arxtandemMinus1<-rbind(A=c(0,1,1,0,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0,0,0,0),  
 G=c(0,0,0,0,0,0,0,0,0,0),  
 T=c(1,0,0,1,1,0,0,1,1,0))  
arxJolma<-rbind( A=c(0,1,1,0,0,0.25,1,1,0,0,1),   
 C=c(0,0,0,0,0,0.25,0,0,0,0,0),  
 G=c(0,0,0,0,0,0.25,0,0,0,0,0),  
 T=c(1,0,0,1,1,0.25,0,0,1,1,0))  
arxTandemNoSpace<-rbind( A=c(0,1,1,0,0,1,0,1,1,0,0,1),  
 C=c(0,0,0,0,0,0,0,0,0,0,0,0),  
 G=c(0,0,0,0,0,0,0,0,0,0,0,0) ,  
 T=c(1,0,0,1,1,0,1,0,0,1,1,0))  
  
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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

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))

## Warning in .Method(..., deparse.level = deparse.level): number of columns  
## of result is not a multiple of vector length (arg 2)

### Match each model to the genome

This code chunk matches each model to the genome

## Match each PWM to the genome  
grangeJolmaMinus<-  
 matchPWM(arxJolma, genome, "100%")  
grangeplaindromicMinus1 <-  
 matchPWM(ArxPlaindrmicMinus1, genome, "100%")  
grangeplaindromicNospace <-  
 matchPWM(arx6MerPWMNospace, genome, "100%")  
grangeplaindromic1space <-  
 matchPWM(arx6MerPWM1space, genome, "100%")  
grangeplaindromic2space <-  
 matchPWM(arx6MerPWM2space, genome, "100%")  
grangeplaindromic3space <-  
 matchPWM(arx6MerPWM3space, genome, "100%")  
grangeplaindromic4space <-  
 matchPWM(arx6MerPWM4space, genome, "100%")  
grangeplaindromic5space <-  
 matchPWM(arx6MerPWM5space, genome, "100%")  
grangeplaindromic6space <-  
 matchPWM(arx6MerPWM6space, genome, "100%")  
grangeplaindromic7space <-  
 matchPWM(arx6MerPWM7space, genome, "100%")  
  
grangeTandemMinusOne <-  
 matchPWM(arxtandemMinus1, genome, "100%")  
grangeTandemNoSpace<-  
 matchPWM(arxTandemNoSpace, genome, "100%")  
grangeTandem1space <-  
 matchPWM(arxTandem1Space, genome, "100%")  
grangeTandem2space <-  
 matchPWM(arxTandem2Space, genome, "100%")  
grangeTandem3space <-  
 matchPWM(arxTandem3Space, genome, "100%")  
grangeTandem4space <-  
 matchPWM(arxTandem4Space, genome, "100%")  
grangeTandem5space <-  
 matchPWM(arxTandem5Space, genome, "100%")  
grangeTandem6space <-  
 matchPWM(arxTandem6Space, genome, "100%")  
grangeTandem7space <-  
 matchPWM(arxTandem7Space, genome, "100%")  
  
#grangeplaindromic1space<-matchPWM(arx6MerPWM1space, genome, "90%")  
#grangeplaindromic2space<-matchPWM(arx6MerPWM2space, genome, "90%")  
#grangeplaindromic3space<-matchPWM(arx6MerPWM3space, genome, "90%")  
#grangeplaindromic4space<-matchPWM(arx6MerPWM4space, genome, "90%")  
#grangeplaindromic5space<-matchPWM(arx6MerPWM4space, genome, "90%")  
#grangeplaindromic6space<-matchPWM(arx6MerPWM4space, genome, "90%")  
#grangeplaindromic7space<-matchPWM(arx6MerPWM4space, genome, "90%")  
#grangeTandem1space<-matchPWM(arxTandem1Space, genome, "90%")  
#grangeTandem2space<-matchPWM(arxTandem2Space, genome, "90%")  
#grangeTandem3space<-matchPWM(arxTandem3Space, genome, "90%")  
#grangeTandem4space<-matchPWM(arxTandem4Space, genome, "90%")  
#grangeTandem5space<-matchPWM(arxTandem5Space, genome, "90%")  
#grangeTandem6space<-matchPWM(arxTandem6Space, genome, "90%")  
#grangeTandem7space<-matchPWM(arxTandem7Space, genome, "90%")

## Code to generate the Plots

1.This code will generate the tables showing the total number of motifs, the number in each genomic location and a bargraph showing that distribution 2.It will also generate a denisty plot showing the proportion of motifs from TSS and another frequency poly showing the number of motifs from TSS.

These graphs will likely also be distorted consequnece of saving these images at higher resoltuions where the Text (therefore the axis and titles) don't scale so we have made them extremely large. and then save it.

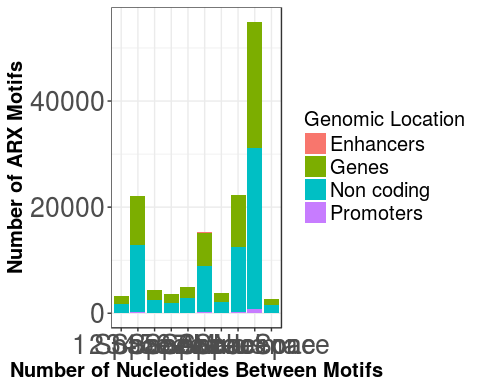
# Generate a data table of the number of each motif model in each genomic region  
tandemDataTable <- rbind(  
 cbind(  
 length(grangeJolmaMinus),  
 sum(countOverlaps(grangeJolmaMinus, UCSCgenes)),  
 sum(countOverlaps(grangeJolmaMinus, promoters)),  
 sum(countOverlaps(grangeJolmaMinus, enhancerGrange)),  
 (length(grangeJolmaMinus)-sum(countOverlaps(grangeJolmaMinus, enhancerGrange))-  
 sum(countOverlaps(grangeJolmaMinus, promoters))- sum(countOverlaps(grangeJolmaMinus, UCSCgenes)))  
),  
 cbind(  
 numberofTandem <- length(grangeTandemMinusOne),  
 dataTableNoGenesminus1 <-  
 sum(countOverlaps(grangeTandemMinusOne, UCSCgenes)),  
 dataTableMinus1 <-  
 sum(countOverlaps(grangeTandemMinusOne, promoters)),  
 dataTableMinus1r <-  
 sum(countOverlaps(grangeTandemMinusOne, enhancerGrange)),  
 (length(grangeTandemMinusOne)-sum(countOverlaps(grangeTandemMinusOne, enhancerGrange))-  
 sum(countOverlaps(grangeTandemMinusOne, promoters))- sum(countOverlaps(grangeTandemMinusOne, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandemNoSpaceSites <- length(grangeTandemNoSpace),  
 dataTableNoGenes <-  
 sum(countOverlaps(grangeTandemNoSpace, UCSCgenes)),  
 dataTableNoSpacePromoters <-  
 sum(countOverlaps(grangeTandemNoSpace, promoters)),  
 dataTableNoSpaceEnhancer <-  
 sum(countOverlaps(grangeTandemNoSpace, enhancerGrange)),  
 (length(grangeTandemNoSpace)-sum(countOverlaps(grangeTandemNoSpace, enhancerGrange))-  
 sum(countOverlaps(grangeTandemNoSpace, promoters))- sum(countOverlaps(grangeTandemNoSpace, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem1spaceSites <- length(grangeTandem1space),  
 dataTable1SpaceGenes <-  
 sum(countOverlaps(grangeTandem1space, UCSCgenes)),  
 dataTable1SpacePromoters <-  
 sum(countOverlaps(grangeTandem1space, promoters)),  
 dataTable1SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem1space, enhancerGrange)),  
 (length(grangeTandem1space)-sum(countOverlaps(grangeTandem1space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem1space, promoters))- sum(countOverlaps(grangeTandem1space, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem2spaceSites <- length(grangeTandem2space),  
 dataTable2SpaceGenes <-  
 sum(countOverlaps(grangeTandem2space, UCSCgenes)),  
 dataTable2SpacePromoters <-  
 sum(countOverlaps(grangeTandem2space, promoters)),  
 dataTable2SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem2space, enhancerGrange)),  
 (length(grangeTandem2space)-sum(countOverlaps(grangeTandem2space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem2space, promoters))- sum(countOverlaps(grangeTandem2space, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem3spaceSites <- length(grangeTandem3space),  
 dataTable3SpaceGenes <-  
 sum(countOverlaps(grangeTandem3space, UCSCgenes)),  
 dataTable3SpacePromoters <-  
 sum(countOverlaps(grangeTandem3space, promoters)),  
 dataTable3SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem3space, enhancerGrange)),  
 (length(grangeTandem3space)-sum(countOverlaps(grangeTandem3space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem3space, promoters))- sum(countOverlaps(grangeTandem3space, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem4spaceSites <- length(grangeTandem4space),  
 dataTable4SpaceGenes <-  
 sum(countOverlaps(grangeTandem4space, UCSCgenes)),  
 dataTable4SpacePromoters <-  
 sum(countOverlaps(grangeTandem4space, promoters)),  
 dataTable4SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem4space, enhancerGrange)),  
 (length(grangeTandem4space)-sum(countOverlaps(grangeTandem4space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem4space, promoters))- sum(countOverlaps(grangeTandem4space, UCSCgenes)))  
 ),   
cbind(  
 numberofTandem5spaceSites <- length(grangeTandem5space),  
 dataTable5SpaceGenes <-  
 sum(countOverlaps(grangeTandem5space, UCSCgenes)),  
 dataTable5SpacePromoters <-  
 sum(countOverlaps(grangeTandem5space, promoters)),  
 dataTable5SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem5space, enhancerGrange)),  
 (length(grangeTandem5space)-sum(countOverlaps(grangeTandem5space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem5space, promoters))- sum(countOverlaps(grangeTandem5space, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem6spaceSites <- length(grangeTandem6space),  
 dataTable6SpaceGenes <-  
 sum(countOverlaps(grangeTandem6space, UCSCgenes)),  
 dataTable6SpacePromoters <-  
 sum(countOverlaps(grangeTandem6space, promoters)),  
 dataTable6SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem6space, enhancerGrange)),  
 (length(grangeTandem6space)-sum(countOverlaps(grangeTandem6space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem6space, promoters))- sum(countOverlaps(grangeTandem6space, UCSCgenes)))  
 ),  
 cbind(  
 numberofTandem7spaceSites <- length(grangeTandem7space),  
 dataTable7SpaceGenes <-  
 sum(countOverlaps(grangeTandem7space, UCSCgenes)),  
 dataTable7SpacePromoters <-  
 sum(countOverlaps(grangeTandem7space, promoters)),  
 dataTable7SpaceEnhancer <-  
 sum(countOverlaps(grangeTandem7space, enhancerGrange)),  
 (length(grangeTandem7space)-sum(countOverlaps(grangeTandem7space, enhancerGrange))-  
 sum(countOverlaps(grangeTandem7space, promoters))- sum(countOverlaps(grangeTandem7space, UCSCgenes)))  
 )  
) %>% as.data.frame  
  
# Colnames and rownames  
colnames(tandemDataTable) <- c("Total",  
 "Motifs in genes",  
 "Motifs in promoters",  
 "Motifs in enhancers",  
 "Non Coding")  
  
  
rownames(tandemDataTable) <- c("Arx Jolma",  
 "Minus one",  
 "No Space",  
 "1 Space",  
 "2 Space",  
 "3 Space",  
 "4 Space",  
 "5 Space",  
 "6 Space",  
 "7 Space")  
tandemDataTable %>% pander()

Table continues below

|  |  |  |  |
| --- | --- | --- | --- |
|  | Total | Motifs in genes | Motifs in promoters |
| **Arx Jolma** | 22346 | 9792 | 263 |
| **Minus one** | 54920 | 23770 | 737 |
| **No Space** | 2626 | 1150 | 39 |
| **1 Space** | 3306 | 1536 | 45 |
| **2 Space** | 22174 | 9343 | 250 |
| **3 Space** | 4400 | 1810 | 56 |
| **4 Space** | 3634 | 1706 | 49 |
| **5 Space** | 4950 | 2066 | 96 |
| **6 Space** | 15222 | 6374 | 159 |
| **7 Space** | 3758 | 1659 | 44 |

|  |  |  |
| --- | --- | --- |
|  | Motifs in enhancers | Non Coding |
| **Arx Jolma** | 44 | 12247 |
| **Minus one** | 96 | 30317 |
| **No Space** | 4 | 1433 |
| **1 Space** | 12 | 1713 |
| **2 Space** | 24 | 12557 |
| **3 Space** | 2 | 2532 |
| **4 Space** | 16 | 1863 |
| **5 Space** | 12 | 2776 |
| **6 Space** | 16 | 8673 |
| **7 Space** | 4 | 2051 |

# write.table(tandemDataTable, file = "~/Thesis/OrientationTandemHumanTable",  
# quote=FALSE,  
# append=FALSE,  
# sep= "\t")  
#   
# write.table(tandemDataTable, file = "~/Thesis/OrientationTandemMouseTable",  
# quote=FALSE,  
# append=FALSE,  
# sep= "\t")  
  
  
##################################  
## Reshape and plot  
##################################  
tandemDataTable <- rownames\_to\_column(tandemDataTable)  
reshapedTandemDataTable<-reshape(tandemDataTable,  
 varying = c( "Motifs in promoters", "Motifs in enhancers", "Non Coding", "Motifs in genes"),  
 v.names = "Numbers of Motif",  
 timevar = "Location",  
 times = c( "Promoters", "Enhancers", "Non coding","Genes" ),  
 direction = "long")  
ggplot(reshapedTandemDataTable, aes(x = rowname, y = `Numbers of Motif`, fill = `Location`)) +  
 geom\_bar(stat = "identity") +  
 xlab(label= "Number of Nucleotides Between Motifs")+  
 ylab(label= "Number of ARX Motifs")+  
 guides(fill=guide\_legend(title="Genomic Location"))+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=20),  
 legend.text = element\_text(size=15),   
 legend.title = element\_text(size=15))+  
 scale\_color\_manual(values=c(`Enhancer`="#999999", `Genes`="#E69F00", `Non-coding`="#56B4E9", `Promoters`= "#56B4E9"))



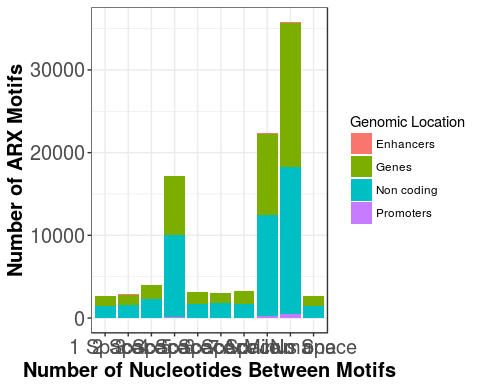
# Plaindromic Data table  
planindromicDataTable <- rbind(  
 cbind(  
 length(grangeJolmaMinus),  
 sum(countOverlaps(grangeJolmaMinus, UCSCgenes)),  
 sum(countOverlaps(grangeJolmaMinus, promoters)),  
 sum(countOverlaps(grangeJolmaMinus, enhancerGrange)),  
 (length(grangeJolmaMinus)-sum(countOverlaps(grangeJolmaMinus, enhancerGrange))-  
 sum(countOverlaps(grangeJolmaMinus, promoters))- sum(countOverlaps(grangeJolmaMinus, UCSCgenes)))  
 ),  
 cbind(  
 length(grangeplaindromicMinus1),  
 sum(countOverlaps(grangeplaindromicMinus1, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromicMinus1, promoters)),  
 sum(countOverlaps(grangeplaindromicMinus1, enhancerGrange)),  
 (length(grangeplaindromicMinus1)-sum(countOverlaps(grangeplaindromicMinus1, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromicMinus1, promoters))- sum(countOverlaps(grangeplaindromicMinus1, UCSCgenes)))  
 ),  
 cbind(  
 length(grangeplaindromicNospace),  
 sum(countOverlaps(grangeplaindromicNospace, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromicNospace, promoters)),  
 sum(countOverlaps(grangeplaindromicNospace, enhancerGrange)),  
 (length(grangeplaindromicNospace)-sum(countOverlaps(grangeplaindromicNospace, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromicNospace, promoters))- sum(countOverlaps(grangeplaindromicNospace, UCSCgenes)))  
 ),  
 cbind(  
 length(grangeplaindromic1space),  
 Arx6mer <- sum(countOverlaps(grangeplaindromic1space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic1space, promoters)),  
 sum(countOverlaps(grangeplaindromic1space, enhancerGrange)),  
 (length(grangeplaindromic1space)-sum(countOverlaps(grangeplaindromic1space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic1space, promoters))- sum(countOverlaps(grangeplaindromic1space, UCSCgenes)))  
 ),  
 cbind(  
 length(grangeplaindromic2space),  
 sum(countOverlaps(grangeplaindromic2space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic2space, promoters)),  
 sum(countOverlaps(grangeplaindromic2space, enhancerGrange)),  
 (length(grangeplaindromic2space)-sum(countOverlaps(grangeplaindromic2space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic2space, promoters))- sum(countOverlaps(grangeplaindromic2space, UCSCgenes)))  
 )  
 ,  
 cbind(  
 numberOfArxSitesPlaindromic3Space <- length(grangeplaindromic3space),  
 sum(countOverlaps(grangeplaindromic3space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic3space, promoters)),  
 sum(countOverlaps(grangeplaindromic4space, enhancerGrange)),  
 (length(grangeplaindromic3space)-sum(countOverlaps(grangeplaindromic3space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic3space, promoters))- sum(countOverlaps(grangeplaindromic3space, UCSCgenes)))  
 ),  
 cbind(  
 numberOfArxSitesPlaindromic4Space <- length(grangeplaindromic4space),  
 sum(countOverlaps(grangeplaindromic4space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic4space, promoters)),  
 sum(countOverlaps(grangeplaindromic4space, enhancerGrange)),  
 (length(grangeplaindromic4space)-sum(countOverlaps(grangeplaindromic4space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic4space, promoters))- sum(countOverlaps(grangeplaindromic4space, UCSCgenes)))  
 ),  
 cbind(  
 numberOfArxSitesPlaindromic5Space <- length(grangeplaindromic5space),  
 sum(countOverlaps(grangeplaindromic5space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic5space, promoters)),  
 sum(countOverlaps(grangeplaindromic5space, enhancerGrange)),  
 (length(grangeplaindromic5space)-sum(countOverlaps(grangeplaindromic5space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic5space, promoters))- sum(countOverlaps(grangeplaindromic5space, UCSCgenes)))  
 ),  
 cbind(  
 numberOfArxSitesPlaindromic6Space <- length(grangeplaindromic6space),  
 sum(countOverlaps(grangeplaindromic6space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic6space, promoters)),  
 sum(countOverlaps(grangeplaindromic6space, enhancerGrange)),  
 (length(grangeplaindromic6space)-sum(countOverlaps(grangeplaindromic6space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic6space, promoters))- sum(countOverlaps(grangeplaindromic6space, UCSCgenes)))  
 ),  
 cbind(  
 numberOfArxSitesPlaindromic7Space <- length(grangeplaindromic7space),  
 sum(countOverlaps(grangeplaindromic7space, UCSCgenes)),  
 sum(countOverlaps(grangeplaindromic7space, promoters)),  
 sum(countOverlaps(grangeplaindromic7space, enhancerGrange)),  
 (length(grangeplaindromic7space)-sum(countOverlaps(grangeplaindromic7space, enhancerGrange))-  
 sum(countOverlaps(grangeplaindromic7space, promoters))- sum(countOverlaps(grangeplaindromic7space, UCSCgenes)))  
 )  
) %>% as.data.frame()  
  
  
  
## Row names and colnames  
colnames(planindromicDataTable) <- c("Total",  
 "Motifs in genes",  
 "Motifs in Promoters",  
 "Motifs in Enhancers",  
 "Non Coding")  
rownames(planindromicDataTable) <-c("Arx Jolma",  
 "Minus one",  
 "No Space",  
 "1 Space",  
 "2 Space",  
 "3 Space",  
 "4 Space",  
 "5 Space",  
 "6 Space",  
 "7 Space")  
  
  
planindromicDataTable %>% pander()

Table continues below

|  |  |  |  |
| --- | --- | --- | --- |
|  | Total | Motifs in genes | Motifs in Promoters |
| **Arx Jolma** | 22346 | 9792 | 263 |
| **Minus one** | 35783 | 17418 | 497 |
| **No Space** | 2626 | 1150 | 39 |
| **1 Space** | 2648 | 1155 | 40 |
| **2 Space** | 2846 | 1244 | 32 |
| **3 Space** | 3952 | 1704 | 45 |
| **4 Space** | 17196 | 7193 | 179 |
| **5 Space** | 3198 | 1496 | 40 |
| **6 Space** | 3052 | 1291 | 50 |
| **7 Space** | 3247 | 1485 | 45 |

|  |  |  |
| --- | --- | --- |
|  | Motifs in Enhancers | Non Coding |
| **Arx Jolma** | 44 | 12247 |
| **Minus one** | 60 | 17808 |
| **No Space** | 4 | 1433 |
| **1 Space** | 4 | 1449 |
| **2 Space** | 7 | 1563 |
| **3 Space** | 15 | 2199 |
| **4 Space** | 15 | 9809 |
| **5 Space** | 8 | 1654 |
| **6 Space** | 4 | 1707 |
| **7 Space** | 7 | 1710 |

# write.table(planindromicDataTable, file = "~/Thesis/OrientationPlaindromicHumanTable",  
# append=FALSE,  
# quote=FALSE,   
# sep="\t")  
#   
# write.table(planindromicDataTable, file = "~/Thesis/OrientationPlaindromicMouseTable",  
# append=FALSE,  
# quote=FALSE,  
# sep="\t")  
  
#Reshape and plot  
planindromicDataTable<- rownames\_to\_column(planindromicDataTable)  
  
reshapedPlaindromicDataTable<-reshape(planindromicDataTable,  
 varying = c( "Motifs in Promoters", "Motifs in Enhancers", "Non Coding", "Motifs in genes"),  
 v.names = "Numbers of Motif",  
 timevar = "Location",  
 times = c( "Promoters", "Enhancers", "Non coding","Genes" ),  
 direction = "long")  
ggplot(reshapedPlaindromicDataTable, aes(x = rowname, y = `Numbers of Motif`, fill = `Location`)) +  
 geom\_bar(stat = "identity") +  
 xlab(label= "Number of Nucleotides Between Motifs")+  
 ylab(label= "Number of ARX Motifs")+  
 guides(fill=guide\_legend(title="Genomic Location"))+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=15) )+  
 scale\_color\_manual(values=c(`Enhancer`="#999999", `Genes`="#E69F00", `Non-coding`="#56B4E9", `Promoters`= "#56B4E9"))



### making frequency plots of each motif model  
  
dataFrameDistance1SpacePromoter <-  
 distanceToNearest(grangeTandem1space, startSites) %>%   
 as.data.frame()  
dataFrameDistance2SpacePromoter <-  
 distanceToNearest(grangeTandem2space, startSites) %>%  
 as.data.frame()  
dataFrameDistance3SpacePromoter <-  
 distanceToNearest(grangeTandem3space, startSites) %>%  
 as.data.frame()  
dataFrameDistance6SpacePromoter <-  
 distanceToNearest(grangeTandem6space, startSites) %>%  
 as.data.frame()  
  
  
dataFrameMerger<-function(z,x,c,v){  
   
 test<-merge(z[3],x[3],by=0, all=TRUE, row.names=NULL)  
 test2<-merge(test, c[3], by=0, all=TRUE, row.names=NULL)  
 test3<- merge(test2, v[3], by=0,all=TRUE, row.names=NULL)  
   
   
 return(test3)  
}  
  
dataFrameDistanceofTandemMotifsFromPromoter<-dataFrameMerger(dataFrameDistance1SpacePromoter,  
 dataFrameDistance2SpacePromoter,   
 dataFrameDistance3SpacePromoter,   
 dataFrameDistance6SpacePromoter)

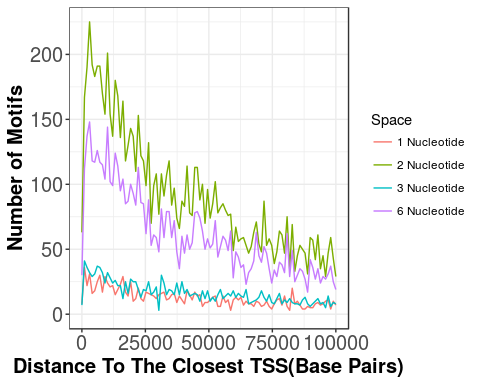
## Warning in merge.data.frame(test, c[3], by = 0, all = TRUE, row.names =  
## NULL): column name 'Row.names' is duplicated in the result

## Warning in merge.data.frame(test2, v[3], by = 0, all = TRUE, row.names =  
## NULL): column names 'Row.names', 'Row.names', 'distance.x', 'distance.y'  
## are duplicated in the result

dataFrameDistanceofTandemMotifsFromPromoter<- dataFrameDistanceofTandemMotifsFromPromoter[4:7]  
colnames(dataFrameDistanceofTandemMotifsFromPromoter)<- c("1 Space",  
 "2 Space",  
 "3 Space",  
 "6 Space")  
ggplotdataFrameDistanceofTandemicMotifsFromPromoter<-reshape(dataFrameDistanceofTandemMotifsFromPromoter,  
 varying = c("1 Space", "2 Space", "3 Space", "6 Space"),  
 v.names = "Distance",  
 timevar = "Space",  
 times = c("1 Nucleotide", "2 Nucleotide", "3 Nucleotide", "6 Nucleotide"),  
 direction = "long")  
  
ggplot(ggplotdataFrameDistanceofTandemicMotifsFromPromoter, aes(x=Distance, group=Space, color=Space))+  
 geom\_freqpoly(bins = 100)+  
 xlab(label = "Distance To The Closest TSS(Base Pairs)")+  
 ylab(label= "Number of Motifs")+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=15) )+  
 scale\_x\_continuous(limits = c(0, 100000))

## Warning: Removed 70526 rows containing non-finite values (stat\_bin).

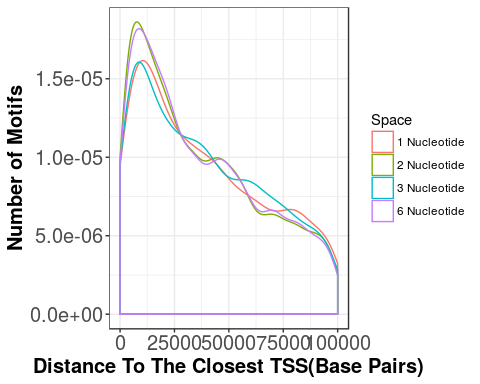
## Warning: Removed 8 rows containing missing values (geom\_path).



ggplot(ggplotdataFrameDistanceofTandemicMotifsFromPromoter, aes(x=Distance, group=Space, color=Space))+  
 geom\_density(bins = 100)+  
 xlab(label = "Distance To The Closest TSS(Base Pairs)")+  
 ylab(label= "Number of Motifs")+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=15) )+  
 scale\_x\_continuous(limits = c(0, 100000))

## Warning: Ignoring unknown parameters: bins

## Warning: Removed 70526 rows containing non-finite values (stat\_density).



##histogram of distances of Plaindromic Motifs  
  
  
dataFrameDistancePlandromic1SpacePromoter <-  
 distanceToNearest(grangeplaindromic1space, startSites) %>% as.data.frame  
dataFrameDistancePlandromic2SpacePromoter <-  
 distanceToNearest(grangeplaindromic2space, startSites) %>% as.data.frame  
dataFrameDistancePlandromic3SpacePromoter <-  
 distanceToNearest(grangeplaindromic3space, startSites) %>% as.data.frame  
dataFrameDistancePlandromic4SpacePromoter <-  
 distanceToNearest(grangeplaindromic4space, startSites) %>% as.data.frame  
  
  
dataFrameDistanceofPlandromicMotifsFromPromoter <- dataFrameMerger(dataFrameDistancePlandromic1SpacePromoter,  
 dataFrameDistancePlandromic2SpacePromoter,  
 dataFrameDistancePlandromic3SpacePromoter,  
 dataFrameDistancePlandromic4SpacePromoter)

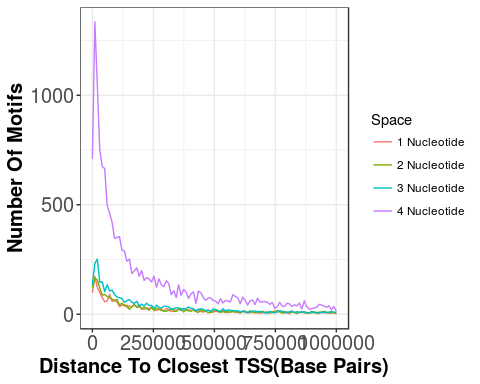
## Warning in merge.data.frame(test, c[3], by = 0, all = TRUE, row.names =  
## NULL): column name 'Row.names' is duplicated in the result

## Warning in merge.data.frame(test2, v[3], by = 0, all = TRUE, row.names =  
## NULL): column names 'Row.names', 'Row.names', 'distance.x', 'distance.y'  
## are duplicated in the result

dataFrameDistanceofPlandromicMotifsFromPromoter<-dataFrameDistanceofPlandromicMotifsFromPromoter[4:7]  
colnames(dataFrameDistanceofPlandromicMotifsFromPromoter)<- c("1 Space",  
 "2 Space",  
 "3 Space",  
 "4 Space")  
  
  
ggplotdataFrameDistanceofPlaindromicMotifsFromPromoter<-reshape(dataFrameDistanceofPlandromicMotifsFromPromoter,  
 varying = c("1 Space", "2 Space", "3 Space", "4 Space"),  
 v.names = "Distance",  
 timevar = "Space",  
 times = c("1 Nucleotide", "2 Nucleotide", "3 Nucleotide", "4 Nucleotide"),  
 direction = "long")  
ggplotdataFrameDistanceofPlaindromicMotifsFromPromoter$Distance<-ggplotdataFrameDistanceofPlaindromicMotifsFromPromoter$Distance%>%as.character%>%as.numeric()  
ggplot(ggplotdataFrameDistanceofPlaindromicMotifsFromPromoter, aes(x=Distance, group=Space, color=Space))+  
 geom\_freqpoly(bins = 100)+  
 xlab(label = "Distance To Closest TSS(Base Pairs)")+  
 ylab(label= "Number Of Motifs")+  
 scale\_x\_continuous(limits = c(0, 1000000))+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=15))

## Warning: Removed 45287 rows containing non-finite values (stat\_bin).

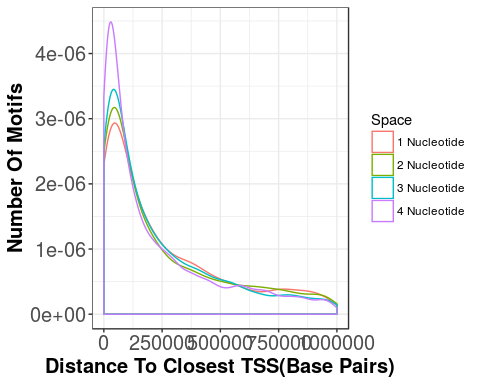
## Warning: Removed 8 rows containing missing values (geom\_path).



ggplot(ggplotdataFrameDistanceofPlaindromicMotifsFromPromoter, aes(x=Distance, group=Space, color=Space))+  
 geom\_density(bins = 100)+  
 xlab(label = "Distance To Closest TSS(Base Pairs)")+  
 ylab(label= "Number Of Motifs")+  
 scale\_x\_continuous(limits = c(0, 1000000))+  
 theme\_bw()+  
 theme(axis.title = element\_text(size=15, face = "bold"),  
 axis.text =element\_text(size=15))

## Warning: Ignoring unknown parameters: bins

## Warning: Removed 45287 rows containing non-finite values (stat\_density).



## Average distances  
Numeric<-apply(dataFrameDistanceofTandemMotifsFromPromoter, 2, as.numeric)  
Numeric<-apply(dataFrameDistanceofPlandromicMotifsFromPromoter, 2, as.numeric)  
Space1Av<-sum(na.omit(Numeric[,1]))/length(na.omit(Numeric[,1]))  
Space2Av<-sum(na.omit(Numeric[,2]))/length(na.omit(Numeric[,2]))  
Space3Av<-sum(na.omit(Numeric[,3]))/length(na.omit(Numeric[,3]))  
Space4Av<-sum(na.omit(Numeric[,4]))/length(na.omit(Numeric[,4]))

sessionInfo()

## R version 3.3.2 (2016-10-31)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 16.04.2 LTS  
##   
## locale:  
## [1] LC\_CTYPE=en\_US.UTF-8 LC\_NUMERIC=C   
## [3] LC\_TIME=en\_US.UTF-8 LC\_COLLATE=en\_US.UTF-8   
## [5] LC\_MONETARY=en\_US.UTF-8 LC\_MESSAGES=en\_US.UTF-8   
## [7] LC\_PAPER=en\_US.UTF-8 LC\_NAME=C   
## [9] LC\_ADDRESS=C LC\_TELEPHONE=C   
## [11] LC\_MEASUREMENT=en\_US.UTF-8 LC\_IDENTIFICATION=C   
##   
## attached base packages:  
## [1] parallel stats4 stats graphics grDevices utils datasets   
## [8] methods base   
##   
## other attached packages:  
## [1] BSgenome.Mmusculus.UCSC.mm9\_1.4.0 BSgenome.Hsapiens.UCSC.hg19\_1.4.0  
## [3] BSgenome\_1.42.0 rtracklayer\_1.34.2   
## [5] MotifDb\_1.16.1 Biostrings\_2.42.1   
## [7] XVector\_0.14.1 plyr\_1.8.4   
## [9] reshape2\_1.4.2 pander\_0.6.1   
## [11] tibble\_1.3.4 ggplot2\_2.2.1   
## [13] GenomicRanges\_1.26.4 GenomeInfoDb\_1.10.3   
## [15] IRanges\_2.8.2 S4Vectors\_0.12.2   
## [17] BiocGenerics\_0.20.0 magrittr\_1.5   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.12 bitops\_1.0-6   
## [3] tools\_3.3.2 zlibbioc\_1.20.0   
## [5] digest\_0.6.12 lattice\_0.20-35   
## [7] evaluate\_0.10.1 gtable\_0.2.0   
## [9] rlang\_0.1.2 Matrix\_1.2-11   
## [11] yaml\_2.1.14 stringr\_1.2.0   
## [13] knitr\_1.17 rprojroot\_1.2   
## [15] grid\_3.3.2 Biobase\_2.34.0   
## [17] XML\_3.98-1.9 BiocParallel\_1.8.2   
## [19] rmarkdown\_1.6 backports\_1.1.0   
## [21] scales\_0.5.0 Rsamtools\_1.26.2   
## [23] htmltools\_0.3.6 GenomicAlignments\_1.10.1   
## [25] SummarizedExperiment\_1.4.0 colorspace\_1.3-2   
## [27] labeling\_0.3 stringi\_1.1.5   
## [29] RCurl\_1.95-4.8 lazyeval\_0.2.0   
## [31] munsell\_0.4.3