Normalization Example using internal CTCF peaks as a standard

Andrew Holding

8/7/2017

Introduction

This is a normalisation using only DeSeq Size Factors to CTCF of MCF7 cells chip treameant etc.

Load convience functions

These functions facilitate the normalisation of data.

```
## Loading required package: GenomicRanges
## Warning: package 'GenomicRanges' was built under R version 3.3.3
## 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
## Warning: package 'S4Vectors' was built under R version 3.3.3
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:base':
##
```

```
##     colMeans, colSums, expand.grid, rowMeans, rowSums

## Loading required package: IRanges

## Warning: package 'IRanges' was built under R version 3.3.3

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## 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")'.

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

##

## Warning: package 'Rsamtools' was built under R version 3.3.3

## Loading required package: Biostrings

## Loading required package: XVector

## Warning: package 'XVector' was built under R version 3.3.3

## <anonymous>: local variable 'treatment_fit' assigned but may not be used
```

Apply settings

```
jg.controlMinOverlap      <- 5
jg.controlSampleSheet     <- "samplesheet/samplesheet_SLX14438_hs_CTCF_DBA.csv"
jg.experimentSampleSheet  <- "samplesheet/samplesheet_SLX14438_hs_ER_DBA.csv"
```

Load control and experimental DiffBind object

To keep file size down these are provided as a Rdata File rather than as raw counts.

```
filename<-"Rdata/example_001_SLX-14438_dba_human_ER_CTCF.rda"
if(!file.exists(filename)){
  dbaExperiment <- jg.getDba(jg.experimentSampleSheet, bRemoveDuplicates=TRUE)
  dbaControl    <- jg.getDba(jg.controlSampleSheet, bRemoveDuplicates=TRUE)
  save(dbaExperiment,dbaControl,file=filename)
} else {
  load(filename)
}
```

Diffbind was simply used as a convient way to extract peak counts. The actual DESeq Pipeline starts simply needs a matrix so we covert from Diffbind and save as csv to provide a starting point for a DESeq pipeline.

```
filename<-"csv/experimentalPeakset.csv"
if(!file.exists(filename)){
  ## Extract Peak set from DiffBind
  jg.experimentPeakset <- jg.dbaGetPeakset(dbaExperiment)
  #Convert Peakset to DeSeq Workflow
```

```
  jg.experimentPeaksetDeSeq<-jg.convertPeakset(jg.experimentPeakset)
  #Save to CSV

  write.csv(jg.experimentPeaksetDeSeq,file=filename)
}

#Repeat for control samples.

filename<-"csv/controlPeakset.csv"
if(!file.exists(filename)){
jg.controlPeakset    <- jg.dbaGetPeakset(dbaControl)
jg.controlPeaksetDeSeq<-jg.convertPeakset(jg.controlPeakset)
write.csv(jg.controlPeaksetDeSeq,   file=filename)
}
```

Pipe line starts here

Load CSV dataset in DESeq format

```
jg.controlPeaksetDeSeq <- read.csv( file="csv/controlPeakset.csv"
                                    ,check.names=FALSE, row.names = 1)
jg.experimentPeaksetDeSeq<- read.csv(file="csv/experimentalPeakset.csv"
                                    ,check.names=FALSE, row.names = 1)
```

```
#Establish size factors directly from Control data
jg.controlSizeFactors = estimateSizeFactorsForMatrix(jg.controlPeaksetDeSeq)

#Get conditions dataframe for DeSeq
jg.conditions <- read.csv(file=jg.controlSampleSheet, header=TRUE, sep=",")['Condition']

#Run DeSeq on control
jg.controlDeSeq<-jg.runDeSeq(jg.controlPeaksetDeSeq, jg.conditions,jg.controlSizeFactors)
```

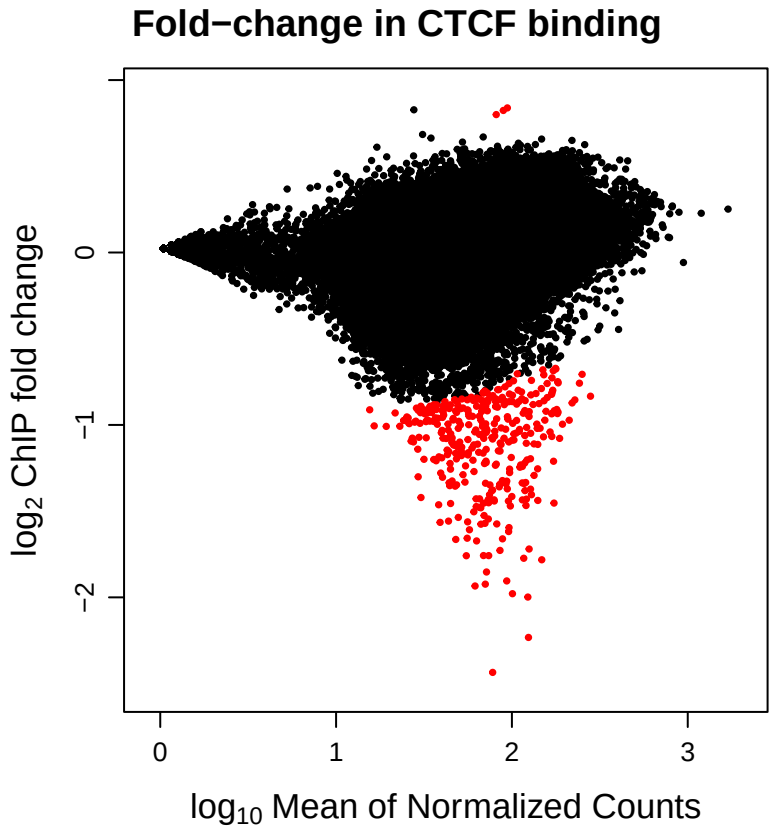
```
## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates
```

```
jg.controlResultsDeseq   = results(jg.controlDeSeq)

#Run DeSeq on experiment
jg.experimentDeSeq<-jg.runDeSeq(jg.experimentPeaksetDeSeq, jg.conditions,jg.controlSizeFactors)
```

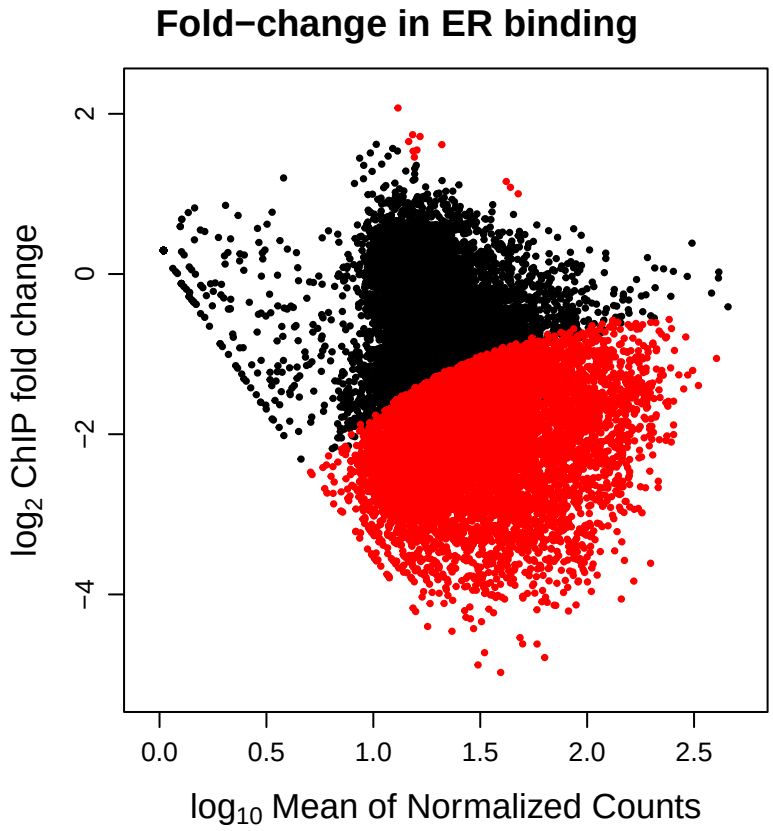
```
## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates
```

```
jg.experimentResultsDeseq   = results(jg.experimentDeSeq)

#Plot results

  jg.plotDeSeq(jg.controlResultsDeseq,
```

```
              p=0.01,
              title.main="Fold-change in CTCF binding",
              flip=T
              )
```




```
jg.plotDeSeq(jg.experimentResultsDeseq,
             p=0.01,
             title.main="Fold-change in ER binding",
             flip=T
             )
```




```
#Draw Combined figure.
  jg.plotDeSeqCombined(jg.controlResultsDeseq,
                       jg.experimentResultsDeseq,
                       title.main="ER and CTCF Binding Folding changes on ER treatment",
                       p=0.01,flip=TRUE)
```

ER and CTCF Binding Folding changes on ER treatment