

Normalization Example using internal CTCF peaks as a standard

Andrew Holding

8/7/2017

Introduction

This is a normalisation using linear regression 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"
jg.treatedCondition      = "Fulvestrant"
jg.untreatedCondition    = "none"

```

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.getDb(a(jg.experimentSampleSheet, bRemoveDuplicates=TRUE)
  dbaControl   <- jg.getDb(a(jg.controlSampleSheet, bRemoveDuplicates=TRUE)
  save(dbaExperiment, dbaControl, file=filename)
} else {
  load(filename)
}

#Load Sample Ids from control sample sheet.
jg.sampleIds <- jg.getSampleIds(jg.controlSampleSheet)

## Extract Peak set from DiffBind
jg.experimentPeakset <- jg.dbaGetPeakset(dbaExperiment)
jg.controlPeakset    <- jg.dbaGetPeakset(dbaControl)

```

```

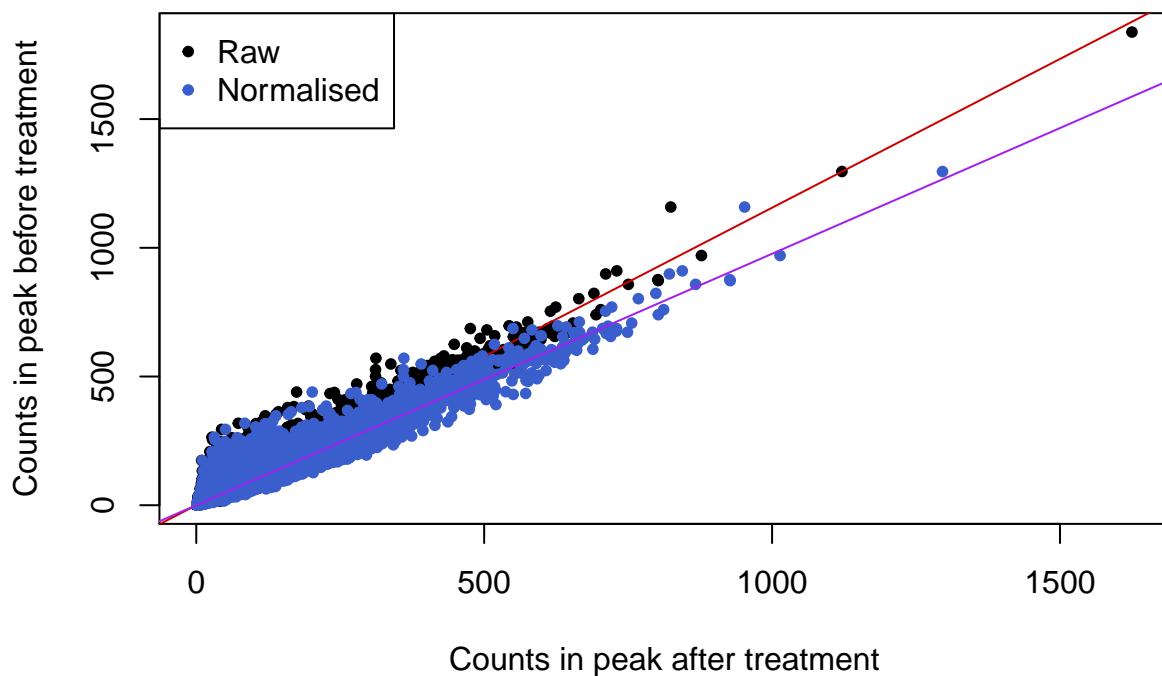
#Get counts for each condition
jg.controlCountsTreated<-jg.getControlCounts(jg.controlPeakset,
                                              jg.controlSampleSheet,
                                              jg.treatedCondition)
jg.controlCountsUntreated<-jg.getControlCounts(jg.controlPeakset,
                                              jg.controlSampleSheet,
                                              jg.untreatedCondition)

#Get sample names for conditions
jg.untreatedNames <- names(jg.controlCountsUntreated)
jg.treatedNames    <- names(jg.controlCountsTreated)

##Plot showing normalization calculation (Optional)
jg.plotNormalization(jg.controlCountsTreated,
                     jg.controlCountsUntreated)

```

Comparision of Counts in peaks



```

## rowMeans(jg.controlCountsTreated)
##                               1.155732

##Get Normalization Coefficient
jg.coefficient<-jg.getNormalizationCoefficient(jg.controlCountsTreated,
                                                jg.controlCountsUntreated)
jg.correctionFactor<-jg.getCorrectionFactor(jg.experimentSampleSheet,
                                             jg.treatedNames,
                                             jg.untreatedNames)

##Apply coefficient and control factor
jg.experimentPeaksetNormalised<-jg.applyNormalisation(jg.experimentPeakset,
                                                       jg.coefficient,
                                                       jg.correctionFactor,
                                                       jg.treatedNames)

```

```

#Return values to Diffbind and plot normalised result.
jg.dba <- DiffBind:::pv.resetCounts(dbaExperiment,
                                      jg.experimentPeaksetNormalised)

jg.dba_analysis<-dba.analyze(jg.dba)

## converting counts to integer mode
## [1] "ANH"

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

dba.plotMA(jg.dba_analysis,bFlip=TRUE)

```

Binding Affinity: Fulvestrant vs. none (10649 FDR < 0.050)

