Package 'ChIPComp'

October 18, 2022

Type Package

Title Quantitative comparison of multiple ChIP-seq datasets

Version 1.26.0

Author Hao Wu, Li Chen, Zhaohui S.Qin, Chi Wang

Maintainer Li Chen <li.chen@emory.edu>

Depends R (>= 3.2.0),GenomicRanges,IRanges,rtracklayer,GenomeInfoDb,S4Vectors

Imports Rsamtools,limma,BSgenome.Hsapiens.UCSC.hg19, BSgenome.Mmusculus.UCSC.mm9,BiocGenerics

Suggests BiocStyle,RUnit

Description ChIPComp detects differentially bound sharp binding sites across multiple conditions considering matching control.

License GPL

LazyLoad yes

biocViews ChIPSeq, Sequencing, Transcription, Genetics,Coverage, MultipleComparison, DataImport

NeedsCompilation yes

git_url https://git.bioconductor.org/packages/ChIPComp

git_branch RELEASE_3_15

git_last_commit 7449627

git_last_commit_date 2022-04-26

Date/Publication 2022-10-18

R topics documented:

`ChIPComp-package` *Detect differential binding sites for ChIP sequencing data*

Description

ChIPComp is an R library performing the differential binding analysis for ChIP-seq count data. Compared with other similar packages (DBChIP, DIME), ChIPComp considers the control samples in the process of detecting the differential binding sites. Extensive simulation results showed that ChIPComp performs favorably compared to DBChIP and DIME when the control samples are ignored. ChIPComp only works for two group comparison at this time, that is, to detect the differential binding sites for one transcription factor(histone) between two conditions (cell lines). We plan to extend the functionalities and make it work for more general experimental designs in the near future.

Author(s)

Hao Wu <hao.wu@emory.edu>, Li Chen <li.chen@emory.edu>

`ChIPComp` *Perform hypothesis testing to detect differential binding sites*

Description

Perform hypothesis testing to detect differential binding sites

Usage

```
ChIPComp(countSet,A,threshold=1)
```

Arguments

`countSet`	A `ChIPComp` object.
`A`	User-specified regions to fit the model. It is a bed file with three columns, named ("chr","start","end"), could be separated by space or tab.
`threshold`	User specified posterior probability threshold. Default is 1.

Value

A object `ChIPComp` contains Column `chr`,`start`,`end` are the binding site genomic coordinate; Column `ip_c(\#condition)_r(\#replicate)` indicates ChIP counts in \#replicate in \#condition; Column `ct_c(\#condition)_r(\#replicate)` indicates smoothing control counts in \#replicate in \#condition; Column `commonPeak` 1s indicate common binding sites; Column `prob.post` is the posterior probability for each binding site. Column `pvalue.wald` is the pvalue of wald test for each binding site.

Author(s)

Hao Wu<hao.wu@emory.edu>, Li Chen <li.chen@emory.edu>

Examples

```
data(seqData)
seqData=ChIPComp(seqData)
```

`makeConf` *make configurations for experimental design written in csv sheet*

Description

Make a list with two elements. The first element is a data frame containing two group comparison study information. The second element is the design matrix.

Usage

```
makeConf(sampleSheet)
```

Arguments

`sampleSheet`	A csv sheet represents ChIP experiments design. It contains 6 columns,`sampleID`,`condition`,`factor`,`ipR` `condition` refers to treatment condition or cell line; `factor` refers to transcription factor or histone modification; `ipReads` is the ChIP sequence data in bam or bed format; `ctReads` is the control sequence data in bam or bed format; `peaks` is the called peaks from existing peak-calling software.

Value

A list with two elements. The first element is a data frame containing two group comparison study information. The second element is the design matrix.

Author(s)

Hao Wu<hao.wu@emory.edu>, Li Chen <li.chen@emory.edu>

Examples

```
confs=makeConf(system.file("extdata", "conf.csv", package="ChIPComp"))
conf=confs$conf
design=confs$design
```

makeCountSet *make differential binding sites data frame*

Description

This is an utility function to create a data frame. The data frame contains binding sites merged by peaks from two conditions, count ChIP read counts, smoothing control counts for each candidate region, and indicate the common peaks from two conditions.

Usage

```
makeCountSet(conf,design,filetype,species,peak.center=FALSE,peak.ext=0,binsize=50,mva.span=c(1000,5
```

Arguments

conf	A data frame that represents the ChIP experiments information. It contains 6 columns,sampleID,condition,factor,ipReads,ctReads,peaks. condition refers to treatment condition or cell line; factor refers to transcription factor or histone modification; ipReads is the ChIP sequence data in bam or bed format; ctReads is the control sequence data in bam or bed format; peaks is the called peaks from existing peak-calling software.
design	Two column design matrix. The number of rows equals number of ChIP samples from two conditions. The first column are all 1s, which indicates intercept in regression model. The second column are 1s for one condition and 0s for another condition.
filetype	Two sequence file types are supported (bed or bam).
species	Two species are supported (hg19 or mm9). Other species are supported by specifying other.
peak.center	This argument is coupled with peak.ext. Default is FALSE. The argument is used when centered regions of peaks are more of interest.
peak.ext	This argument is coupled with peak.center. Default is 0.
binsize	binsize in bp to calculate the smooth local lambda in poisson distribution. The default is 50bp.
mva.span	1 kb, 5 kb or 10 kb window centered at the peak location in the control sample.

Value

A object ChIPComp. Column chr,start,end are the binding site genomic coordinate; Column ip_c(\#condition)_r(\#replicate) indicates the ChIP counts in \#replicate in \#condition; Column ct_c(\#condition)_r(\#replicate) indicates the smoothing control counts in \#replicate in \#condition; Column commonPeak indicates the common binding sites.

Examples

```
conf=data.frame(
SampleID=1:4,
condition=c("Helas3","Helas3","K562","K562"),
factor=c("H3k27ac","H3k27ac","H3k27ac","H3k27ac"),
ipReads=system.file("extdata",c("Helas3.ip1.bed","Helas3.ip2.bed","K562.ip1.bed","K562.ip2.bed"),package="ChIP
ctReads=system.file("extdata",c("Helas3.ct.bed","Helas3.ct.bed","K562.ct.bed","K562.ct.bed"),package="ChIPComp
peaks=system.file("extdata",c("Helas3.peak.bed","Helas3.peak.bed","K562.peak.bed","K562.peak.bed"),package="Ch
)
conf$condition=factor(conf$condition)
  conf$factor=factor(conf$factor)
design=as.data.frame(lapply(conf[,c("condition","factor")],as.numeric))-1
design=as.data.frame(model.matrix(~condition,design))
countSet=makeCountSet(conf,design,filetype="bed", species="hg19",binsize=1000)
```

`plot.ChIPComp` *plot correlation between log ChIP read counts and smoothing control counts in common binding sites.*

Description

plot correlation between log ChIP counts and smoothing control counts in common binding sites.

Usage

```
## S3 method for class 'ChIPComp'
plot(x,...)
```

Arguments

`x`	A `ChIPComp` object.
`...`	Other graphical parameters to `plot`

Value

Plot the correlation between ChIP sample and control sample

Author(s)

Hao Wu<hao.wu@emory.edu>, Li Chen <li.chen@emory.edu>

Examples

```
data(seqData)
plot(seqData)
```

print.ChIPComp *Print top ranked differential binding sites*

Description

Print top differential binding sites ranked by posterior probability in a decreasing order.

Usage

```
## S3 method for class 'ChIPComp'
print(x,topK=10,...)
```

Arguments

x	A ChIPComp object.
topK	top K differential binding sites. Default is 10.
...	Other parameters to print

Value

Print differential binding sites ranked by posterior probability

Author(s)

Hao Wu<hao.wu@emory.edu>, Li Chen <li.chen@emory.edu>

Examples

```
data(seqData)
seqData=ChIPComp(seqData)
print(seqData)
```

seqData *A* ChIPComp *object.*

Description

The object is sampled from 50 common binding sites between Helas3 and K562 cell lines for H3K27ac and 5 unique binding sites for each cell line.

Usage

```
data(seqData)
```

Value

A "ChIPComp" class object

Examples

```
data(seqData)
```

Index