Package 'DBChIP'

October 16, 2019

Type Package

Title Differential Binding of Transcription Factor with ChIP-seq

Version 1.28.0

Date 2012-06-21

Author Kun Liang

Maintainer Kun Liang <kliang@stat.wisc.edu>

Depends R (>= 2.15.0), edgeR, DESeq

Suggests ShortRead, BiocGenerics

Description DBChIP detects differentially bound sharp binding sites across multiple conditions, with or without matching control samples.

License GPL (>= 2)

LazyLoad yes

biocViews ChIPSeq, Sequencing, Transcription, Genetics

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

git_branch RELEASE_3_9

git_last_commit 33b7b1d

git_last_commit_date 2019-05-02

Date/Publication 2019-10-15

R topics documented:

`DBChIP-package` *DBChIP-package*

Description

Detecting differential binding of transcription factors with ChIP-seq

Details

Package:	DBChIP
Type:	Package
Version:	1.1
Date:	2011-09-26
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Kun Liang

Maintainer: Kun Liang <kliang@stat.wisc.edu>

References

Liang, K and Keles, S (2012) *Detecting differential binding of transcription factors with ChIP-seq*, 28, 121-122.

See Also

`DBChIP`

Examples

```
data("PHA4")
dat <- DBChIP(binding.site.list, chip.data.list=chip.data.list, input.data.list=input.data.list, conds=conds
rept <- report.peak(dat)
rept
```

`binding.site.list` *Binding site predictions*

Description

Binding site predictions for emb and L1 conditions in a study of transcription factor PHA-4/FOXA in *C.elegans*

Usage

```
data("PHA4")
```

Format

List of 2 elements: emb and L1. Each element is a data.frame with fields: chr, strand and weight (optional).

Details

Weight represents a measure of strength of the binding, for example, the number of reads in the peak.

Source

Zhong et al. (2010), *Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response*, PLoS Genetics, 6, e1000848.

See Also

PHA4

Examples

```
data(PHA4)
names(binding.site.list)
head(binding.site.list[["emb"]])
```

---

chip.data.list *ChIP data of transcription factor PHA-4/FOXA in* C.elegans

---

Description

This data set contains parts of ChIP-seq data of transcription factor PHA-4/FOXA in *C.elegans*. Only ChIP data in chromosome I with position < 0.9M bp are included.

Usage

```
data("PHA4")
```

Format

List of 4 elements: "emb_rep1", "emb_rep2", "L1_rep1" and "L1_rep2". Each element is a data.frame with fields: chr, strand and pos.

Source

Zhong et al. (2010), *Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response*, PLoS Genetics, 6, e1000848.

See Also

PHA4

Examples

```
data(PHA4)
names(chip.data.list)
head(chip.data.list[["emb_rep1"]])
```

---

conds *conds*

Description

Experimental conditions of ChIP replicates in PHA4 data.

Usage

```
data("PHA4")
```

Format

The format is: Factor w/ 2 levels "emb","L1": 1 1 2 2

Details

The first two ChIP replicates are in embryonic (emb) condition, and the last two are in the first stage of larval development (L1) condition.

Source

Zhong et al. (2010), *Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response*, PLoS Genetics, 6, e1000848.

See Also

PHA4

Examples

```
data(PHA4)
conds
```

DBChIP *DBChIP*

Description

Detecting differential binding of transcription factors with ChIP-seq

Usage

```
DBChIP(binding.site.list, chip.data.list, conds, input.data.list = NULL,
 data.type = c("MCS", "AlignedRead", "BED"), frag.len = 200, chr.vec = NULL,
 chr.exclusion = NULL, chr.len.vec = NULL, subtract.input = FALSE, norm.factor.vec = NULL,
 in.distance = 100, out.distance = 250, window.size = 250,
 dispersion=NULL, common.disp=TRUE, prior.n=10,
 two.sample.method="composite.null", allowable.FC=1.5, collapsed.quant=0.5)
```

Arguments

binding.site.list	a list of data.frames. Each data.frame corresponds to one condition in comparison and has three fields, chr, pos, and weight, to indicate the binding location and strength.
chip.data.list	a list of ChIP data where each list item corresponds to one ChIP library. The name of the items should be unique. Biological replicates should be in separate items. Each item can be one of three accepted data types: MCS, AlignedRead and BED.
conds	a vector of conditions of ChIP libraries. Should be the same order as chip.data.list, or the names should be specified as a permutation of the names of chip.data.list.
input.data.list	a list of control data. Should have same data type as in chip.data.list. The names of the items should be unique, and each name should be matched to either a ChIP replicate name when the two are paired or a condition name in general.
data.type	"MCS", "AlignedRead" or "BED". See Details.
frag.len	average fragment length. Default 200 bp.
chr.vec	a vector of chromosomes in data. User can specify chr.vec, or it can be computed from the ChIP and control samples.
chr.exclusion	user can either specify chr.vec, or specify the chromosomes to exclude through this parameter.
chr.len.vec	a vector of chromosome lengths corresponding to chr.vec. Can be specified if known, or will be computed as the largest 5' end position in the data.
subtract.input	logical. Whether input will be subtracted from ChIP when counting the binding reads. Default is FALSE.
norm.factor.vec	a vector of normalization factors between the ChIP and control libraries when controls are available. Can be specified, or will be computed by DBChIP.
in.distance	within cluster distance. If the distance between centroids of two clusters are smaller than this value, the clusters will be merged into one. Default value 100 bp.

`out.distance` out of cluster distance. If the distance between centroids of two clusters are larger than this value, they are considered different binding sites. Also double-function as the window size to count reads around each site. Default value 250 bp.

`window.size` the window size to count reads around a binding site.

`dispersion` The dispersion parameter in Negative Binomial distribution. Could be a numerical value or a vector with a length of the number of consensus sites.

`common.disp` logical, TRUE (use common dispersion parameter for all sites) or FALSE (use site-specific dispersion).

`prior.n` a parameter regulate the degree of pooling when using site-specific dispersion (`common.disp`=FALSE).

`two.sample.method` the method to use when comparing two condition with no replicates. The default is to test a composite null that allow certain fold change `allowable.FC`. Otherwise user should provide a dispersion parameter.

`allowable.FC` allowable fold change when testing a composite null. Default value 1.5.

`collapsed.quant` the quantile to use when testing more than two conditions without replicates. Default value is 0.5, the median.

Details

The ChIP and control data should be properly filtered before the analysis to avoid artifacts. For example, reads mapping to mitochondrial DNA, or Y chromosome for female samples will need to be filtered.

Filtering of chromosomes can be achieved through specification of `chr.vec` and/or `chr.exclusion`. Only reads from chromosomes in `chr.vec` but not in `chr.exclusion` are utilized in the analysis.

User can include or exclude sex chromosomes in the computation, depending on whether protein-DNA bindings on sex chromosomes are of research interest.

Biological replicates of a ChIP sample should be kept separate so that dispersion can be properly estimated. On the other hand, replicates of a control/input sample should be merged because the purpose of the control samples is to estimate the background for testing and plotting. One exception would be when a control replicate is paired with a ChIP replicate, for example, they are coming from the same batch, a portion of which is used for IP and the other portion is used for control. In such case, the control replicate can be kept separate with the same name of the matching ChIP replicate.

data.type

- `MCS` Minimum ChIP-Seq format. data.frame with fields: chr (factor), pos (integer) and strand (factor, "+" and "-"). pos is 5' location. This is different from eland default which use 3' location for reverse strand.
- `AlignedRead` from Bioconductor ShortRead package (with support of commonly used formats, including Eland, MAQ, Bowtie, SOAP and BAM).
- `BED` with at least first 6 fields (chrom, start, end, name, score and strand), http://genome.ucsc.edu/FAQ/FAQformat.ht

Users are recommended to study the histogram of the p-values for model checking. More specifically, the p-values between 0.5 and 1 should be roughly uniform. When many replicates are available, users can also randomly split biological replicates of the same condition and perform

comparisons through DBChIP using the estimated dispersion parameter to check whether the p-values look uniform.

Value

A list with following components:

`chip.list`	the list of ChIP data in internal MCS format.
`conds`	vector of conditions of ChIP libraries.
`frag.len`	average fragment length.
`chr.vec`	a vector of chromosomes in data.
`chr.len.vec`	a vector of chromosome lengths corresponding to chr.vec.
`consensus.site`	consensus sites. It is a data.frame, for details, see `site.merge`.
`site.count`	a data.frame of read counts at each consensus site.
`test.stat`	a data.frame of test statistics for testing non-differential binding at each site, include p-values and fold changes.
`input.list`	the list of control data. The components from this and below are only available when control data are available.
`matching.input.names`	the matching input names for ChIP libraries.
`norm.factor.vec`	vector of normalization factors between the ChIP and control libraries when controls are available. Can be specified, or will be computed by DBChIP.

Author(s)

Kun Liang, <kliang@stat.wisc.edu>

References

Liang, K and Keles, S (2012). Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics*, 28, 121-122.

See Also

`DBChIP-package`, `PHA4`, `read.binding.site.list`, `site.merge`.

Examples

```
data("PHA4")
dat <- DBChIP(binding.site.list, chip.data.list=chip.data.list, input.data.list=input.data.list, conds=conds
rept <- report.peak(dat)
rept
#pdf("Diff.Binding.pdf")
plotPeak(rept, dat)
#dev.off()

## experienced users can proceed in a step by step fashion such that if program
## needs to be run for a different setting, intermediate results can be saved and reused.
data("PHA4")
conds <- factor(c("emb","emb","L1", "L1"), levels=c("emb", "L1"))
```

```
bs.list <- read.binding.site.list(binding.site.list)

## compute consensus site
consensus.site <- site.merge(bs.list, in.distance=100, out.distance=250)

dat <- load.data(chip.data.list=chip.data.list, conds=conds, consensus.site=consensus.site, input.data.list=

## count ChIP reads around each binding site
dat <- get.site.count(dat, window.size=250)

## test for differential binding
dat <- test.diff.binding(dat)

# report test result and plot the coverage profiles
rept <- report.peak(dat)
rept
plotPeak(rept, dat)
```

`get.site.count` *Get site count*

Description

Count number of reads around each binding site

Usage

```
get.site.count(dat, subtract.input=FALSE, window.size = 250)
```

Arguments

`dat`	a list with the following items: consensus.site, chip.list, input.list, matching.input.names, norm.factor.vec. Description of the items can be found in the return value of DBChIP.
`subtract.input`	logical. Whether input will be subtracted from ChIP when counting the binding reads. Default is FALSE.
`window.size`	the window size to count reads around each binding site. Default 250 bp.

Details

The read count for each binding site is defined as the sum of the number of 5' ends on the positive strand within the upstream window `[s-w,s-1]` and the number of 5' ends on the negative strand within the downstream window `[s+1,s+w]`, where `s` is a consensus site position and `w` is the window size parameter.

Value

This function return the incoming `dat` with the new component:

`site.count`	a matrix of read counts for each site (row) and each ChIP library (column).

See Also

DBChIP

input.data.list *Control/input data of transcription factor PHA-4/FOXA in* C.elegans

Description

This data set contains parts of ChIP-seq data of transcription factor PHA-4/FOXA in *C.elegans*. Only control data in chromosome I with position < 0.9M bp are included.

Usage

```
data("PHA4")
```

Format

List of 2 elements: "emb" and "L1". Each element is a data.frame with fields: chr, strand and pos.

Source

Zhong et al. (2010), *Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response*, PLoS Genetics, 6, e1000848.

See Also

PHA4

Examples

```
data(PHA4)
names(input.data.list)
head(input.data.list[["emb"]])
```

load.data *load.data*

Description

Assemble data for further analysis.

Usage

```
load.data(chip.data.list, conds, consensus.site, input.data.list = NULL, data.type = "MCS", chr.vec
```

Arguments

chip.data.list	a list of ChIP data where each list item corresponds to one ChIP library. The name of the items should be unique. Biological replicates should be in separate items. Each item can be one of three accepted data types: MCS, AlignedRead and BED.
conds	a vector of conditions of ChIP libraries. Should be the same order as chip.data.list, or the names should be specified as a permutation of the names of chip.data.list.
consensus.site	consensus binding sites. Should be the result of site.merge.
input.data.list	a list of control data. Should have same data type as in chip.data.list. The names of the items should be unique, and each name should be matched to either a ChIP replicate name when the two are paired or a condition name in general.
data.type	"MCS", "AlignedRead" or "BED". See Details.
chr.vec	a vector of chromosomes in data. User can specify chr.vec, or it can be computed from the ChIP and control samples.
chr.exclusion	user can either specify chr.vec, or specify the chromosomes to exclude through this parameter.
chr.len.vec	a vector of chromosome lengths corresponding to chr.vec. Can be specified if known, or will be computed as the largest 5' end position in the data.
norm.factor.vec	a vector of normalization factors between the ChIP and control libraries when controls are available. Can be specified, or will be computed by DBChIP.
frag.len	average fragment length. Default 200 bp.

Details

The ChIP and control data should be properly filtered before the analysis to avoid artifacts. For example, reads mapping to mitochondrial DNA, or Y chromosome for female samples will need to be filtered.

Filtering of chromosomes can be achieved through specification of chr.vec and/or chr.exclusion. Only reads from chromosomes in chr.vec but not in chr.exclusion are utilized in the analysis.

User can include or exclude sex chromosomes in the computation, depending on whether protein-DNA bindings on sex chromosomes are of research interest.

Biological replicates of a ChIP sample should be kept separate so that dispersion can be properly estimated. On the other hand, replicates of a control/input sample should be merged because the purpose of the control samples is to estimate the background for testing and plotting. One exception would be when a control replicate is paired with a ChIP replicate, for example, they are coming from the same batch, a portion of which is used for IP and the other portion is used for control. In such case, the control replicate can be kept separate with the same name of the matching ChIP replicate.

data.type

- MCS Minimum ChIP-Seq format. data.frame with fields: chr (factor), pos (integer) and strand (factor, "+" and "-"). pos is 5' location. This is different from eland default which use 3' location for reverse strand.
- AlignedRead from Bioconductor ShortRead package (with support of commonly used formats, including Eland, MAQ, Bowtie, SOAP and BAM).
- BED with at least first 6 fields (chrom, start, end, name, score and strand), http://genome.ucsc.edu/FAQ/FAQformat.ht

Value

A list with following components:

`chip.list`	the list of ChIP data in internal MCS format.
`conds`	vector of conditions of ChIP libraries.
`frag.len`	average fragment length.
`chr.vec`	a vector of chromosomes in data.
`chr.len.vec`	a vector of chromosome lengths corresponding to chr.vec.
`consensus.site`	consensus sites. It is a data.frame, for details, see `site.merge`.
`input.list`	the list of control data. The components from this and below are only available when control data are available.
`matching.input.names`	the matching input names for ChIP replicates.
`chip.background.size`	the background sizes (number of reads) for ChIP replicates. Background is defined by exluding the neighborhoods of the consensus sites.
`input.background.size`	the background sizes for input replicates.
`norm.factor.vec`	vector of normalization factors between the ChIP and control libraries when controls are available. Can be specified, or will be computed by DBChIP.

See Also

DBChIP.

Examples

```
data("PHA4")
conds <- factor(c("emb","emb","L1", "L1"), levels=c("emb", "L1"))

bs.list <- read.binding.site.list(binding.site.list)

## compute consensus site
consensus.site <- site.merge(bs.list, in.distance=100, out.distance=250)

#load data
dat <- load.data(chip.data.list=chip.data.list, conds=conds, consensus.site=consensus.site, input.data.list=
names(dat)
```

PHA4 *ChIP-seq data of transcription factor PHA-4/FOXA in* C.elegans

---

Description

This data set contains parts of ChIP-seq data of transcription factor PHA-4/FOXA in *C.elegans*. ChIP and control data and identified binding sites in chromosome I with position < 0.9M bp are included.

Usage

```
data("PHA4")
```

Format

PHA4 has three elements:

- `binding.site.list` a list of binding sites for embryonic and L1 conditions
- `chip.data.list` a list of ChIP reads data for embryonic and L1 conditions
- `input.data.list` a list of control reads data for embryonic and L1 conditions
- `conds` a vector of conditions of ChIP libraries.

References

Zhong et al. (2010), *Genome-wide identification of binding sites defines distinct functions for Caenorhabditis elegans PHA-4/FOXA in development and environmental response*, PLoS Genetics, 6, e1000848.

See Also

DBChIP

`plotPeak` *Plot peak*

Description

Plot the coverage profile of differentially bound peaks

Usage

```
plotPeak(rept, dat, lib.size = NULL, w = 400, ext = 200, combine.rep = FALSE, cap = NULL, n.row.per.pa
```

Arguments

`rept`	a data.frame as the result of `report.peak`.
`dat`	a list with the following items: chip.list, input.list, matching.input.names, norm.factor.vec. Description of the items can be found in the return value of DBChIP.
`lib.size`	a vector of library size of each ChIP sample. Used to scale profiles so that they are comparable.
`w`	half window size to plot around the binding sites.
`ext`	extension size for each read. Each read is extended from its 5' end by `ext` towards 3' end. If `ext` is set to 1, the minimum, per nucleotide read counts will be plotted.
`combine.rep`	logical, whether to combine replicates for the plot. Can be useful when there are too many replicates to plot.
`cap`	the maximum number of reads per nucleotide allowd to plot.
`n.row.per.page`	the maximum number of tracks (rows) per page.
`caption`	additional caption to appear on the title besides the location.

See Also

DBChIP

`read.binding.site.list` *read.binding.site.list*

Description

Process the list of binding sites into an internal format

Usage

```
read.binding.site.list(binding.site.list)
```

Arguments

`binding.site.list`
: a list of data.frames. Each data.frame corresponds to one condition in comparison and has three fields, chr, pos and weight, to indicate the binding location and strength.

Details

The binding site for each condition is split into a list where binding sites are grouped according to their chromosomes.

Value

A list of conditions, each of which is a list of binding sites in a certain chromosome. The binding sites are in data.frames with two fields, pos and weight, to indicate the binding location and strength.

See Also

DBChIP

Examples

```
data("PHA4")
bs.list <- read.binding.site.list(binding.site.list)
str(bs.list)
```

`report.peak` *Report peaks*

Description

Report most significant peaks

Usage

```
report.peak(test.res, FDR = NULL, FDR.method = "BH", n = 10, add.origin = TRUE, adaptive.threshold =
```

Arguments

`test.res`	a list with the item `test.stat`, which is a data.frame of test statistics for testing non-differential binding at each site, include p-values and fold changes.
`FDR`	the desireable false discovery rate (FDR) level.
`FDR.method`	the method to control FDR. Default is "BH", the Benjamini and Hochberg (1995) method. Another option is "adaptive", which use the adaptive.threshold to estimate the number of true null hypotheses.
`n`	the top `n` differential peaks to return.
`add.origin`	logical. Whether to add peaks' origin information (how many and which conditions the consensus site are merged from) in the result.
`adaptive.threshold`	vector of two values between 0 and 1. The number of true null hypotheses is estimated as the number of p-values between these two values divided by the distance between these two values. Default is (0.05, 0.95). If it is set to (0.05, 1), it becomes the method recommended in Blanchard and Roquain (2009).

Details

The default is to return the top `n` differential peaks. If user specify `FDR`, a set of peaks under the threshold will be returned instead. The FDR is computed through the classical Benjamini & Hochberg 1995 method.

Value

a data.frame with with following fields

`chr`	chromosome.
`pos`	consensus binding position.
`nsig`	number of significant original binding sites that are merged into consensus site.
`origin`	the origin of merged binding sites.
`ori.pos`	the original positions of merged binding sites. The consensus position is a weighted average of original positions.
`FC.condition name`	the fold change comparing to the first condition.
`pval`	p-value for testing non-differential binding.
`FDR`	the q-value.

References

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B*, 57, 289-300.

Blanchard, G. and Roquain, E. (2009). Adaptive false discovery rate control under independence and dependence. *Journal of Machine Learning Research*, 10, 2837-2871.

See Also

DBChIP

`site.merge` *Cluster close-by sites into consensus sites*

Description

Cluster close-by sites into consensus sites through agglomerative (bottom-up) hierarchical clustering.

Usage

```
site.merge(bs.list, in.distance = 100, out.distance = 250)
```

Arguments

`bs.list`	internal format of binding site. Result of `read.binding.site.list`.
`in.distance`	within cluster distance. If the distance between centroids of two clusters are smaller than this value, the clusters will be merged into one. Default value 100 bp.
`out.distance`	out of cluster distance. If the distance between centroids of two clusters are larger than this value, they are considered different binding sites. Default value 250 bp.

Details

We group predicted locations from multiple conditions into clusters of close-by locations by using agglomerative (bottom-up) hierarchical clustering with centroid linkage.

If the distance between centroids of two clusters are between `in.distance` and `out.distance`, the cluster with higher weight will be kept.

Value

A list where each element represents a chromosome. Within each chromosome, it is a data.frame with following fields

`pos`	consensus binding position.
`nsig`	number of significant original binding sites that are merged into consensus site.
`origin`	the origin of merged binding sites.
`ori.pos`	the original positions of merged binding sites. The consensus position is a weighted average of original positions.

See Also

DBChIP

`test.diff.binding` *Test for differential binding*

Description

A null hypothesis of non-differential binding is tested at each consensus site.

Usage

```
test.diff.binding(dat, lib.size = NULL, dispersion = NULL, common.disp = TRUE, prior.n = 10, two.sam
```

Arguments

`dat`	a list with the item `site.count`, which is a matrix of read counts at binding sites.
`lib.size`	a vector of library size of each ChIP sample.
`dispersion`	The dispersion parameter in Negative Binomial distribution. Could be a numerical value or a vector with a length of the number of consensus sites.
`common.disp`	logical, TRUE (use common dispersion parameter for all sites) or FALSE (use site-specific dispersion).
`prior.n`	a parameter regulate the degree of pooling when using site-specific dispersion (`common.disp`=FALSE).
`two.sample.method`	the method to use when comparing two condition with no replicates. The default is to test a composite null that allow certain fold change `allowable.FC`. Otherwise user should provide a dispersion parameter.
`allowable.FC`	allowable fold change when testing a composite null. Default value 1.5.
`collapsed.quant`	the quantile to use when testing more than two conditions without replicates. Default value is 0.5, the median.

Details

Users are recommended to study the histogram of the p-values for model checking. More specifically, the p-values between 0.5 and 1 should be roughly uniform. When many replicates are available, users can also randomly split biological replicates of the same condition and perform comparisons through DBChIP using the estimated dispersion parameter to check whether the p-values look uniform.

Value

This function return the incoming `dat` with new field

`test.stat`	a data.frame of test statistics for testing non-differential binding at each site, include p-values and fold changes.

Author(s)

Kun Liang, <kliang@stat.wisc.edu>

References

Liang, K and Keles, S (2012) *Detecting differential binding of transcription factors with ChIP-seq*, 28, 121-122.

See Also

DBChIP

Index