Package 'SimBindProfiles'

March 26, 2024

Title Similar Binding Profiles

Version 1.40.0

Date 2013-09-17

Author Bettina Fischer, Enrico Ferrero, Robert Stojnic, Steve Russell

Maintainer Bettina Fischer <bef22@cam.ac.uk>

Description

SimBindProfiles identifies common and unique binding regions in genome tiling array data. This package does not rely on peak calling, but directly compares binding profiles processed on the same array platform. It implements a simple threshold approach, thus allowing retrieval of commonly and differentially bound regions between datasets as well as events of compensation and increased binding.

Depends R (>= 2.10), methods, Ringo

Imports limma, mclust, Biobase

biocViews Microarray, Software

License GPL-3

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

git_branch RELEASE_3_18

git_last_commit cea1be0

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-25

R topics documented:

`SimBindProfiles-package` *Similar Binding Profiles, identifies common and unique regions in array genome tiling array data*

Description

SimBindProfiles, identifies common and unique binding regions in genome tiling array data. This package does not rely on peak calling, but directly compares binding profiles processed on the same array platform. It implements a simple threshold approach, thus allowing retrieving commonly and differentially bound regions between datasets as well as events of compensation and increased binding.

Details

Package:	SimBindProfiles
Type:	Package
License:	GPL-3

Author(s)

Bettina Fischer <bef22@cam.ac.uk> Enrico Ferrero <ef300@cam.ac.uk> Robert Stojnic <rs550@cam.ac.uk> Steve Russell <sr120@cam.ac.uk> Maintainer: Bettina Fischer <bef22@cam.ac.uk>

`compensationRegions` *Classify two Binding Profiles - Compensation*

Description

This function is used to classify three Binding Profiles into regions which are bound in data set one and two but not three.

Usage

```
compensationRegions(xSet, sgrset = c(1, 2, 3), bound.cutoff, diff.cutoff,
  probes, probe.max.spacing, writeBedFile = TRUE)
```

Arguments

xSet	object of class ExpressionSet
sgrset	vector of lenght 3; specifying which data sets to compare from the ExpressionSet
bound.cutoff	numeric; threshold above probes are considered "bound"
diff.cutoff	numeric; difference threshold to determine if objects uniquely bound
probes	integer; minimum number of probes in a valid region
probe.max.spacing	integer; maximum number of base pairs in a gap before splitting a region into 2 regions
writeBedFile	logical; should bed file be written

Details

Select probes with a signal above the bound.cutoff in data set 1 and 2 and below in set 3, and for which the average signal of set 1 and set2 is above the diff.cutoff compared to set 3. These probes are then filtered into regions using the probes and probe.max.spacing details. The score is calculated as mean (probes in region average(set 1 and set 2) minus set 3). Optional bed file formated result files are written using the choosen options in the file names.

Value

data.frame with the following columns:

name	name(s) of data set to which region belongs
class.group	class group, in this case only 1
chr	chromsome
start	start position of region
end	end position of region
scrore	score of region
nProbes	number of probes in region

Author(s)

Bettina Fischer, Robert Stojnic

See Also

pairwiseRegions, threewayRegions, increasedBindingRegions

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
transcompABC <- compensationRegions(SGR, sgrset=c(1,2,3), bound.cutoff=1.86,
                diff.cutoff=1.4, probes=10, probe.max.spacing=200)
```

`eSetScatterPlot` *Scatterplot of ExpressionSet object*

Description

Create a smooth scatterplot with correlation

Usage

```
eSetScatterPlot(xSet)
```

Arguments

`xSet`	object of class ExpressionSet

Details

Scatterplot and correlations of all data sets in ExpressionSet object

Value

No useful return. The function is called for its side effect to produce the scatterplot.

Author(s)

Bettina Fischer

See Also

`plot`, `smoothScatter`

Examples

```
dataPath <- system.file("data", package = "SimBindProfiles")
load(file.path(dataPath, "SGR.RData"))
eSetScatterPlot(SGR)
```

`findBoundCutoff` *Find the bound.cutoff*

Description

Define the bound.cutoff using either the normalNull or the twoGaussiansNull method.

Usage

```
findBoundCutoff(xSet, method = c("normalNull", "twoGaussiansNull"), mean.method = "mode",
   pvalue = FALSE, fdr = FALSE, pvalPlot = FALSE)
```

Arguments

`xSet`	object of class ExpressionSet
`method`	a character string equal to "normalNull" or "twoGaussiansNull"
`mean.method`	a characyer string equal to "mode" or "zero". This is used in combination with the "normalNull" method.
`pvalue`	decimal specifying the p-value cutoff (either pvalue or fdr can be used not both!)
`fdr`	decimal specifying the fdr cutoff (either pvalue or fdr can be used not both!)
`pvalPlot`	logical, if TRUE the pvalue histogram is written

Details

We implemented two methods to set the bound.cutoff, probes above this threshold are considered "bound". The twoGaussiansNull method established in the Ringo package (Toedling et al., 2007), by which the data is assumed to follow a mixture of two Gaussian distributions. The one Gaussian with the lower mean value is assumed to be the null distribution and probe levels are assigned p-values based on this null distribution. Alternatively the user can select the normalNull method instead which assumes the null distribution is normal and symmetrical around the mode (or zero). For both methods the user can decide if the resulting p-values are to be adjusted for multiple testing (fdr) or selected by a p-value threshold. The function also provides QC plots for the twoGaussiansNull and an optional p-value histogram.

Value

Returns a numeric

Note

Please note that the use of the package "mclust" is only free for strict academic use (see the license of "mclust" here: http://www.stat.washington.edu/mclust/license.txt). The alternative function normalNull does not have this restriction.

Author(s)

Bettina Fischer

References

Toedling J., Skylar O., Krueger T, Fischer J.J., Sperling S., Huber W. 2007 Ringo - an R/Bioconductor package for analyzing ChIP-chip readouts. BMC Bioinformatics, 8:221

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
bound.cutoff <- findBoundCutoff(SGR, method="twoGaussiansNull", fdr=0.25)
```

`increasedBindingRegions`

Classify two Binding Profiles - Increased Binding

Description

This function is used to classify two Binding Profiles into regions which are more bound in one data set than the other.

Usage

```
increasedBindingRegions(xSet, sgrset = c(1, 2), bound.cutoff, diff.cutoff, probes,
  probe.max.spacing, writeBedFile = TRUE)
```

Arguments

`xSet`	object of class ExpressionSet
`sgrset`	vector of lenght 2; specifying which data sets to compare from the ExpressionSet
`bound.cutoff`	numeric; threshold above probes are considered "bound"
`diff.cutoff`	numeric; difference threshold to determine if object 1 and object 2 are uniquely bound
`probes`	integer; minimum number of probes in a valid region
`probe.max.spacing`	integer; maximum number of base pairs in a gap before splitting a region into 2 regions
`writeBedFile`	logical; should a bed file be written

Details

Probe signal values above the bound.cutoff threshold for both data sets are compared where set 1 is above the diff.cutoff of set 2. These probes are then filtered into regions using the probes and probe.max.spacing details. The score is calculated as mean (probes in region set 1 minus set 2). Optional bed file formated result files are written using the choosen options in the file names.

Value

data.frame with the following columns:

name	name(s) of data set to which region belongs
class.group	class group, in this case only 1
chr	chromsome
start	start position of region
end	end position of region
scrore	score of region
nProbes	number of probes in region

Author(s)

Bettina Fischer, Robert Stojnic

See Also

pairwiseRegions, compensationRegions, threewayRegions, plotBoundProbes

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
overcompBC <- increasedBindingRegions(SGR, sgrset=c(2,3), bound.cutoff=1.86, diff.cutoff=1.4,
              probes=10, probe.max.spacing=200)
```

---

pairwiseRegions *Classify two Binding Profiles*

Description

Classify two Binding Profiles into unique and common binding regions and write results to bed files.

Usage

```
pairwiseRegions(xSet, sgrset = c(1, 2), bound.cutoff, diff.cutoff, probes, probe.max.spacing, writeBed
```

Arguments

xSet	object of class ExpressionSet
sgrset	vector of lenght 2; specifying which data sets to compare from the ExpressionSet
bound.cutoff	numeric; threshold above probes are considered “bound”
diff.cutoff	numeric; difference threshold to determine if object 1 and object 2 are unqiquely bound

probes	integer; minimum number of probes in a valid region
probe.max.spacing	integer; maximum number of base pairs in a gap before splitting a region into 2 regions
writeBedFile	logical; should bed file be written

Details

Probe signal values above the bound.cutoff in both data are classified as common bound. Probes which are above the bound.cutoff and in one data and higher than the diff.cutoff to the other data are called unique. Then these probes are then filtered into regions using the probes and probe.max.spacing details. The score for the unique regions is calculated as mean (probes in region set 1 minus set 2), or vise versa. The score for the common region is the mean (probes in region (set 1 plus set 2)/2). Optional bed file formated result files are written using the choosen options in the file names.

Value

data.frame with the following columns:

name	name(s) of data set to which region belongs
class.group	class group; 1, 2 or 3 for common regions between both sets
chr	chromsome
start	start position of region
end	end position of region
scrore	score of region
nProbes	number of probes in region

Author(s)

Bettina Fischer, Robert Stojnic

See Also

compensationRegions, increasedBindingRegions, threewayRegions

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
pairAB <- pairwiseRegions(SGR, sgrset=c(1,2), bound.cutoff=1.86, diff.cutoff=1.4,
          probes=10, probe.max.spacing=200)
```

plotBoundProbes *Plot coloured bound probes*

Description

Scatterplot of bound probes in colour based on the method.

Usage

```
plotBoundProbes(xSet, sgrset, method=c("pairwise" , "compensation" , "increasedBinding"),
                bound.cutoff, diff.cutoff, cols=NULL, pcex=2)
```

Arguments

xSet	object of class ExpressionSet
sgrset	integer; specifying which data set to use from the ExpressionSet
method	a character string equal to “pairwise” , “increasedBinding” , “compensation”
bound.cutoff	numeric; threshold above probes are considered “bound”
diff.cutoff	numeric; difference threshold to determine if object 1 and object 2 are uniquely bound
cols	vector of colours to highlight probes, otherwise colours are set by default
pcex	a numerical vector giving the amount by which probe symbols should be scaled relative to the default

Details

Scatterplot of the bound probes in colour based on the selected method

Value

Coloured scatter plot

Author(s)

Bettina Fischer

See Also

plot, pairwiseRegions, compensationRegions, increasedBindingRegions

Examples

```
 dataPath <- system.file("data",package="SimBindProfiles")
 load(file.path(dataPath,"SGR.RData"))
plotBoundProbes(SGR, sgrset=c(1,2), method="pairwise", bound.cutoff=1.86, diff.cutoff=1.4)
```

probeAnnoFromESet *Build a probeAnno from ExpressionSet*

Description

Build a probeAnno from ExpressionSet.

Usage

```
probeAnnoFromESet(eSet, probeLength)
```

Arguments

eSet	object of class ExpressionSet
probeLength	integer; specifying probe lenght on array

Details

Creates a probeAnno object from ExpressionSet object. The function uses the PROBE_ID, CHROMOSOME and POSITION information store in the ExpressionSet object.

Value

An object of class probeAnno holding the mapping between probes and genomic positions.

Author(s)

Bettina Fischer

See Also

probeAnno-class

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
probeAnno <- probeAnnoFromESet(SGR, probeLength=50)
```

probeLengthPlot *Plot the probe length frequency*

Description

Plot the probe length frequency.

Usage

```
probeLengthPlot(xSet, sgrset = 1, chr = NULL, bound.cutoff, probe.max.spacing = 200, xlim.max = 25)
```

Arguments

xSet	object of class ExpressionSet
sgrset	integer; specifying which data set to use from the ExpressionSet
chr	probes from which chromosome to be used, default NULL means all probes
bound.cutoff	numeric; threshold above probes are considered "bound"
probe.max.spacing	integer; maximum amount of base pairs at which bound probes are condensed into one region.
xlim.max	integer; maximum number of probes plotted along the x-axsis

Details

Plot the probe length frequency per regions based on a bound.cutoff value.

Value

Plot of frequency of the number of probes within bound regions

Author(s)

Bettina Fischer

See Also

plot

Examples

```
 dataPath <- system.file("data",package="SimBindProfiles")
 load(file.path(dataPath,"SGR.RData"))
probeLengthPlot(SGR, sgrset=1, chr=NULL, bound.cutoff=1.2, probe.max.spacing=200, xlim.max=50)
```

readSgrFiles *Read sgr data*

Description

Function to read sgr files into a list object, quantile normalise signal.

Usage

```
readSgrFiles(X, dataPath = getwd(), fileExt=".txt", normalise = TRUE)
```

Arguments

X	object of class ExpressionSet
dataPath	path to the directory holding the data sgr data files
normalise	logical, should data be quantil normalised
fileExt	character specifying the filename extension

Details

The function reads the data files in sgr file tab delimited format: chr, position, signal. All data sets must be from the same array platform and have the same chromosome names and positions. The data is read and then quantile normalised and stored as an ExpressionSet.

Value

Returns normalized, transformed values as an object of class ExpressionSet

Author(s)

Bettina Fischer

See Also

ExpressionSet

Examples

```
sgrfiles <- c("SoxNDam_trunc","SoxN-DDam_trunc","DDam_trunc")
dataPath <- system.file("extdata",package="SimBindProfiles")
readTestSGR <- readSgrFiles(X=sgrfiles, dataPath)
```

SGR *A Sample Data object*

Description

The SGR datset consists of three data sets stored in an ExpressionSet object. These are SoxN-Dam, SoxN-DDam and DDam with probes located on chromosome X between 1-6000000 bp. The precomputed dataset contains precomputed objects for the bound.cutoff, compR and increasedR to save time for building the vignette.

Usage

```
data(SGR)
data(precomputed)
```

Format

An ExpressionSet object, with three data sets.

Examples

```
data(SGR)
data(precomputed)
```

threewayRegions *Classify three Binding Profiles*

Description

Classify three Binding Profiles into unique and common binding regions and write results to bed files.

Usage

```
threewayRegions(xSet, sgrset = c(1, 2, 3), bound.cutoff, diff.cutoff,
  probes, probe.max.spacing, writeBedFile=TRUE)
```

Arguments

xSet	object of class ExpressionSet
sgrset	vector of lenght 3; specifying which data sets to compare from the ExpressionSet
bound.cutoff	numeric; threshold above probes are considered “bound”
diff.cutoff	numeric; difference threshold to determine if objects are unqiquely bound
probes	integer; minimum number of probes in a valid region

`probe.max.spacing`
: integer; maximum number of base pairs in a gap before splitting a region into 2 regions

`writeBedFile`
: logical; should a bed file be written

Details

Signal values are flagged as bound and unbound, and then the bound probes are classified into 7 groups: class.group=1: unique probes in object 1; class.group=2: unique probes in object 2; class.group=3: unique probes in object 3; class.group=4: common probes in object 1+2; class.group=5: common probes in object 2+3; class.group=6: common probes in object 1+3; class.group=7: common probes in object 1+2+3. The classified probes are then filtered into regions using the probes and probe.max.spacing details. For scores calculations please refer to the pairwiseRegions function. Optional bed file formated result files are written using the choosen options in the file names.

Value

`name`
: name(s) of data set to which region belongs

`class.group`
: class group; 1 to 7, see details above

`chr`
: chromsome

`start`
: start position of region

`end`
: end position of region

`scrore`
: score of region

`nProbes`
: number of probes in region

Author(s)

Bettina Fischer, Robert Stojnic

See Also

`pairwiseRegions`, `compensationRegions`, `increasedBindingRegions`

Examples

```
dataPath <- system.file("data",package="SimBindProfiles")
load(file.path(dataPath,"SGR.RData"))
threewayRegions(SGR, sgrset=c(1,2,3), bound.cutoff=1.86, diff.cutoff=1.4,
                probes=10, probe.max.spacing=200)
```

Index