Package 'DiffBind'

September 24, 2012

Type Package

Version 1.2.4

Title Differential Binding Analysis of ChIP-Seq peak data

Date 2012-09-07

Author
Rory Stark<rory.stark@cancer.org.uk>, Gordon Brown <gordon.brown@cancer.org.uk>

Maintainer Rory Stark<rory.stark@cancer.org.uk>

Description Compute differentially bound sites from multiple ChIP-seq experiments using affinity (quantitative) data. Also enables occupancy (overlap) analysis and plotting functions.

biocViews Bioinformatics, HighThroughputSequencing, ChIPseq

License Artistic-2.0

LazyLoad yes

Depends R (>= 2.14.0), GenomicRanges

Imports RColorBrewer, amap, edgeR (>= 2.3.58), gplots, limma, DESeq,grDevices, stats, utils, IRanges, zlibbioc

Suggests DESeq

Enhances rgl, parallel

LinkingTo Rsamtools

Collate core.R parallel.R counts.R contrast.R analyze.R io.R helper.R utils.R DBA.R

R topics documented:

`DiffBind-package` *Differential Binding Analysis of ChIP-seq peaksets*

Description

Differential binding analysis of ChIP-seq peaksets

Details

Computes differentially bound sites from multiple ChIP-seq experiments using affinity (quantitative) data. Also enables occupancy (overlap) analysis and plotting functions.

Entry Points:

dba:	Construct a dba object
dba.peakset:	Add a peakset to, or retrieve a peakset from, a dba object
dba.overlap:	Compute binding site overlaps and/or correlations
dba.count:	Count reads in binding sites
dba.contrast:	Establish contrast(s) for analysis
dba.analyze:	Execute affinity analysis
dba.report:	Generate report for a contrast analysis
dba.plotHeatmap:	Heatmap plot
dba.plotPCA:	Principal Components plot
dba.plotBox:	Boxplots
dba.plotMA:	MA/scatter plot
dba.plotVenn:	Venn diagram plot
dba.show:	Show dba metadata
dba.mask:	Mask samples or sites
dba.save:	Save dba object
dba.load:	Load dba object

Author(s)

Rory Stark <rory.stark@cancer.org.uk> and Gordon Brown <gordon.brown@cancer.org.uk>

---

dba *Construct a DBA object*

Description

Constructs a new DBA object from a sample sheet, or based on an existing DBA object

Usage

```
dba(DBA,mask, minOverlap=2,
    sampleSheet="dba_samples.csv",
    config=data.frame(RunParallel=TRUE, reportInit="DBA"),
    caller='raw', skipLines=0, bAddCallerConsensus=FALSE,
    bRemoveM=TRUE, bRemoveRandom=TRUE,
    bCorPlot=FALSE, attributes)
```

Arguments

DBA	existing DBA object – if present, will return a fully-constructed DBA object based on the passed one, using criteria specified in the mask and/or minOverlap parameters. If missing, will create a new DBA object based on the sampleSheet.
mask	logical or numerical vetcor indicating which peaksets to include in the resulting model if basing DBA object on an existing one. See dba.mask.
minOverlap	only include peaks in at least this many peaksets in the main binding matrix if basing DBA object on an existing one.
sampleSheet	data frame containing sample sheet, or file name of sample sheet to load (ignored if DBA is specified). Columns names in sample sheet should include: • SampleID: Identifier string for sample • Tissue: Identifier string for tissue type • Factor: Identifier string for factor • Condition: Identifier string for condition • Replicate: Replicate number of sample • bamReads: file path for bam file containing aligned reads for ChIP sample • bamControl: file path for bam file containing aligned reads for control sample • ControlID: Identifier string for control sample (optional) • Peaks: path for file containing peaks for sample. format determined by PeakCaller field or caller parameter • PeakCaller: Identifier string for peak caller used. If Peaks is not a bed file, this will determine how the Peaks file is parsed. If missing, will use default peak caller specified in caller parameter. Possible values: – "raw": text file file; peak score is in fourth column – "bed": .bed file; peak score is in fifth column – "macs": MACS .xls file

  - "swembl": SWEMBL .peaks file
  - "bayes": bayesPeak file
  - "peakset": peakset written out using pv.writepeakset
  - "fp4": FindPeaks v4

`config` data frame containing sample sheet, or file name of config file to load when constructing a new DBA object from a sample sheet. NULL indicates no config file. Relevant fields include:

- RunParallel: logical indicating if counting and analysis operations should be run in parallel using multicore by default.
- DataType: default class for peaks and reports (DBA_DATA_GRANGES, DBA_DATA_RANGEDDATA, or DBA_DATA_FRAME).
- AnalysisMethod: either DBA_EDGER or DBA_DESEQ.

`caller` if a sampleSheet is specified, the default peak file format that will be used if the PeakCaller column is absent.

`skipLines` if a sampleSheet is specified, the number of lines (ie header lines) at the beginning of each peak file to skip.

`bAddCallerConsensus` add a consensus peakset for each sample with more than one peakset (i.e. different peak callers) when constructing a new DBA object from a sample sheet.

`bRemoveM` logical indicating whether to remove peaks on chrM (mitochondria) when constructing a new DBA object from a sample sheet.

`bRemoveRandom` logical indicating whether to remove peaks on chrN_random when constructing a new DBA object from a sample sheet.

`bCorPlot` logical indicating that a correlation heatmap should be plotted before returning

`attributes` vector of attributes to use subsequently as defaults when generating labels in plotting functions:

- DBA_ID
- DBA_TISSUE
- DBA_FACTOR
- DBA_CONDITION
- DBA_REPLICATE
- DBA_CONSENSUS
- DBA_CALLER
- DBA_CONTROL

Details

MODE: Construct a new DBA object from a samplesheet:

dba(sampleSheet, config, bAddCallerConsensus, bRemoveM, bRemoveRandom, attributes)

MODE: Construct a DBA object based on an existing one:

dba(DBA, mask, attributes)

Value

DBA object

Author(s)

Rory Stark and Gordon Brown

Examples

```
# Create DBA object from a samplesheet
setwd(system.file("extra", package="DiffBind"))
tamoxifen = dba(sampleSheet="tamoxifen.csv")
tamoxifen

#Create a DBA object with a subset of samples
data(tamoxifen_peaks)
Responsive = dba(tamoxifen,tamoxifen$masks$Responsive)
Responsive
```

DBA object methods *Standard S3 methods for DBA object*

Description

Standard S3 methods for DBA object.

Usage

```
## S3 method for class 'DBA'
print(x, ...)
## S3 method for class 'DBA'
summary(object, ...)
## S3 method for class 'DBA'
plot(x, ...)
```

Arguments

x	DBA object
object	DBA object
...	Arguments passed on to parent methods

Details

S3 methods for DBA object

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
tamoxifen
data(tamoxifen_counts)
tamoxifen
```

DBA tamoxifen resistance dataset
Tamoxifen resistance dataset used for DBA examples

Description

Tamoxifen resistance dataset used for DBA examples

Usage

```
data(tamoxifen_peaks)

data(tamoxifen_counts)

data(tamoxifen_analysis)
```

Arguments

tamoxifen_peaks	load tamoxifen resistance dataset DBA object with peak (occupancy) data
tamoxifen_counts	load tamoxifen resistance dataset DBA object with count (affinity) data
tamoxifen_analysis	load tamoxifen resistance dataset DBA object with count (affinity) data and edgeR-based differential binding analysis results

Details

The tamoxifen resistance dataset is used for the DBA vignette and man page examples.

Value

loads a DBA object named tamoxifen

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
tamoxifen
data(tamoxifen_counts)
plot(tamoxifen)
data(tamoxifen_analysis)
dba.plotMA(tamoxifen)
```

dba.analyze *Perform differential binding affinity analysis*

Description

Performs differential binding affinity analysis

Usage

```
dba.analyze(DBA, method=DBA$config$AnalysisMethod,
            bSubControl=TRUE, bFullLibrarySize=FALSE, bTagwise=TRUE,
            bCorPlot=TRUE,  bReduceObjects=T, bParallel=DBA$config$RunParallel)
```

Arguments

DBA	DBA object. If no contrasts are specified (DBA$contrast is NULL), default contrasts will be added via a call to dba.contrast(DBA).
method	method, or vector of methods, by which to analyze differential binding affinity. Supported methods: • DBA_EDGER • DBA_DESEQ • DBA_EDGER_CLASSIC • DBA_DESEQ_CLASSIC • DBA_EDGER_GLM • DBA_DESEQ_GLM
bSubControl	logical indicating whether Control read counts are subtracted for each site in each sample before performing analysis.
bFullLibrarySize	logical indicating if the full library size (total number of reads in BAM/SAM/BED file) for each sample is used for scaling normalization. If FALSE, the total number of reads present in the peaks for each sample is used (generally preferable).
bTagwise	logical indicating if dispersion should be calculated on a tagwise (or per-condition) basis. If there are only a very few members of each group in a contrast (e.g. no replicates), this should be set to FALSE.
bCorPlot	logical indicating whether to plot a correlation heatmap for the analyzed data (first contrast only). If no sites are significantly differentially bound using the default threholds, no heatmap will be plotted.
bReduceObjects	logical indicating whether strip the analysis objects of unnecessary fields to save memory. If it is desired to used the DBA$contrasts[[n]]$edgeR and/or DBA$contrasts[[n]]$DESeq objects directly in the edgeR and/or DESeq packages, this should be set to FALSE.
bParallel	logical indicating that the analyses is to be done in parallel using multicore (one process for each contrast for each method, plus an additional process per method).

Details

See the DBA User Guide for more details on how the edgeR and DESeq analyses are carried out.

Value

DBA object with results of analysis added to DBA$contrasts.

Note

If the "edgeR" method is specified, and there is a blocking factor for the contrast(s) specified using a previous call to dba.contrast, a multi-factor analysis will automatically be carried out in addition to a single factor analysis.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_counts)

tamoxifen = dba.analyze(tamoxifen)
tamoxifen

tamoxifen = dba.analyze(tamoxifen,method=c(DBA_EDGER,DBA_DESEQ))
tamoxifen
```

---

dba.contrast *Set up contrasts for differential binding affinity analysis*

Description

Sets up contrasts for differential binding affinity analysis

Usage

```
dba.contrast(DBA, group1, group2=!group1, name1="group1", name2="group2",
             minMembers=3, block ,
             categories = c(DBA_TISSUE,DBA_FACTOR,DBA_CONDITION,DBA_TREATMENT))
```

Arguments

DBA	DBA object with count data
group1	mask of samples in first group (when adding a specific contrast). See dba.mask
group2	mask of samples in second group (when adding a specific contrast). See dba.mask
name1	label for samples in first group (when adding a specific contrast).
name2	label for samples in second group (when adding a specific contrast).
minMembers	when automatically generating contrasts, minimum number of unique samples in a group. Must be at least 2, as replicates are strongly advised. If you wish to do an analysis with no replicates, you can set the group1 and group2 parameters explicitly.
categories	when automatically generating contrasts, attribute or vector of attributes to base contrasts on:

- DBA_ID
- DBA_TISSUE
- DBA_FACTOR
- DBA_CONDITION
- DBA_TREATMENT
- DBA_REPLICATE
- DBA_CALLER

`block` blocking attribute for multi-factor analysis. This may be specified as either a value, a vector, or a list.

If block is a value, the specified metadata field is used to derive the blocking factor. One of:

- DBA_TISSUE
- DBA_FACTOR
- DBA_CONDITION
- DBA_TREATMENT
- DBA_REPLICATE
- DBA_CALLER

If block is a vector, it can either be a mask (logical vector) or a vector of peakset numbers. In this case, the peaksets indicated in the blocking vector are all given the same value (true), while any peaksets not included in the vector take the alternative value (false).

If block is a list, it should be a list of vectors (either logical masks or vectors of peakset numbers), with each indicating a set of peaksets that should share the same value. Each peasket should appear at most once, and any peaksets not specified will be given an default value (other).

Details

MODE: Set up all possible contrasts:

dba.contrast(DBA, minMembers, categories)

MODE: Set up a specific contrast:

dba.contrast(DBA, group1, group2, name1, name2, block)

Value

DBA object with contrast(s) set as DBA$contrasts. Contrast list can be retrieved using dba.show(DBA, bContrasts=T).

Note

Contrasts will only be set up for peaksets where DBA_CALLER == "counts".

Contrasts can be cleared by DBA$contrasts=NULL.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_counts)
tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION)
tamoxifen

# Another way to do the same thing
tamoxifen$contrasts=NULL
tamoxifen = dba.contrast(tamoxifen, tamoxifen$masks$Responsive, tamoxifen$masks$Resistant,
                                            "Responsive", "Resistant")
tamoxifen

# Add add default contrasts
tamoxifen$contrasts=NULL
tamoxifen = dba.contrast(tamoxifen)
tamoxifen

# Specify a blocking factor
tamoxifen$contrasts=NULL
tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION, block=DBA_TISSUE)
tamoxifen

tamoxifen$contrasts=NULL
tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION, block=list(c(3,4,5,8,9),c(1,2,10,11)))
tamoxifen

tamoxifen$contrasts=NULL
tamoxifen = dba.contrast(tamoxifen, categories=DBA_CONDITION, block=tamoxifen$masks$MCF7)
tamoxifen = dba.analyze(tamoxifen)
tamoxifen
```

---

dba.count *Count reads in binding site intervals*

Description

Counts reads in binding site intervals. Files must be one of bam, bed and gzip-compressed bed. File suffixes must be ".bam", ".bed", or ".bed.gz" respectively.

Usage

```
dba.count(DBA, peaks, minOverlap=2, score=DBA_SCORE_TMM_MINUS_EFFECTIVE, bLog=FALSE,
          insertLength, maxFilter, bRemoveDuplicates=FALSE, bCalledMasks=TRUE,
          bCorPlot=TRUE, bParallel=DBA$config$RunParallel)
```

Arguments

DBA	DBA object
peaks	GRanges, RangedData, dataframe, or matrix containing intervals to use. If missing, generates a consensus peakset using minOverlap parameter. If NULL, changes the score used in the global binding matrix to the score type specified in the score parameter.

`minOverlap` only include peaks in at least this many peaksets when generating consensus peakset (i.e. when peaks parameter is missing).

`score` which score to use in the binding affinity matrix. Note that all raw read counts are maintained for use by dba.analyze, regardless of how this is set. One of:

DBA_SCORE_READS	raw read count for interval using only reads from ChIP
DBA_SCORE_READS_FOLD	raw read count for interval from ChIP divided by read count for interval f
DBA_SCORE_READS_MINUS	raw read count for interval from ChIP minus read count for interval from
DBA_SCORE_RPKM	RPKM for interval using only reads from ChIP
DBA_SCORE_RPKM_FOLD	RPKM for interval from ChIP divided by RPKM for interval from control
DBA_SCORE_TMM_READS_FULL	TMM normalized (using edgeR), using ChIP read counts and Full Library
DBA_SCORE_TMM_READS_EFFECTIVE	TMM normalized (using edgeR), using ChIP read counts and Effective Li
DBA_SCORE_TMM_MINUS_FULL	TMM normalized (using edgeR), using ChIP read counts minus Control r
DBA_SCORE_TMM_MINUS_EFFECTIVE	TMM normalized (using edgeR), using ChIP read counts minus Control r

`bLog` logical indicating whether log2 of score should be used (only applies to DBA_SCORE_RPKM_FOLD and DBA_SCORE_READS_FOLD).

`insertLength` if present, this value will be used as the length of the reads. Each read will be extended from its endpoint along the appropriate strand by this many bases. If missing, the read size indicated in the BAM/SAM/BED file will be used.

`maxFilter` value to use for filtering intervals with low read counts. Only intervals where at least one sample has at least maxFilter reads will be included. If missing, includes all intervals. If peaks is NULL, will remove sites from existing DBA object without recounting.

`bRemoveDuplicates` logical indicating if duplicate reads (ones that map to exactly the same genomic position) should be removed. If TRUE, any location where multiple reads map will be counted as a single read.

`bCalledMasks` logical indicating whether to compute site masks for each peakset indicating which sites were originally identified as peaks(used by dba.report).

`bCorPlot` logical indicating whether to plot a correlation heatmap for the counted data

`bParallel` if TRUE, use multicore to get counts for each read file in parallel

Value

DBA object with binding affinity matrix based on read count scores.

Author(s)

Rory Stark and Gordon Brown

Examples

```
# These won't run unless you have the reads available in a BAM, SAM, or BED file
data(tamoxifen_peaks)
## Not run: tamoxifen = dba.count(tamoxifen)


# Count using a peakset made up of only peaks in all responsive MCF7 replicates
data(tamoxifen_peaks)
mcf7Common = dba.overlap(tamoxifen,tamoxifen$masks$MCF7&tamoxifen$masks$Responsive)
```

```
## Not run: tamoxifen = dba.count(tamoxifen,peaks=mcf7Common$inAll)
tamoxifen

# Change binding affinity scores
data(tamoxifen_counts)
tamoxifen = dba.count(tamoxifen,peaks=NULL,score=DBA_SCORE_READS)
head(tamoxifen$vectors)
tamoxifen = dba.count(tamoxifen,peaks=NULL,score=DBA_SCORE_RPKM_FOLD)
head(tamoxifen$vectors)
tamoxifen = dba.count(tamoxifen,peaks=NULL,score=DBA_SCORE_TMM_MINUS_FULL)
head(tamoxifen$vectors)
```

`dba.load` *load DBA object*

Description

Reads in saved DBA object

Usage

```
dba.load(file='DBA', dir='.', pre='dba_', ext='RData')
```

Arguments

`file`	main filename
`dir`	directory in which to save model
`pre`	string to pre-pend to filename
`ext`	file extension to use

Value

loaded DBA object

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
dba.save(tamoxifen,'tamoxifenPeaks')
tamoxifen = dba.load('tamoxifenPeaks')
```

dba.mask *Derive a mask to define a subset of peaksets or sites for a DBA object*

Description

Derives a mask to define a subset of peaksets or sites for a DBA object.

Usage

```
dba.mask(DBA, attribute, value, combine='or', mask, merge='or', bApply=FALSE,
         peakset, minValue=-1)
```

Arguments

DBA	DBA object
attribute	when deriving a peakset mask, attribute to base mask on: • DBA_ID • DBA_TISSUE • DBA_FACTOR • DBA_CONDITION • DBA_TREATMENT • DBA_REPLICATE • DBA_CONSENSUS • DBA_CALLER • DBA_CONTROL
value	when deriving a peakset/sample mask, attribute value (or vector of attribute values) to match.
combine	when deriving a peakset/sample mask, if value is a vector, OR when deriving a site mask, and peaksets is a vector, this is method for combining result of each value: • "or" • "and" • "nor" • "nand"
mask	when deriving a peakset/sample mask, this specifies an existing mask to merge with; if missing, create new mask
merge	when deriving a peakset/sample mask, and an existing mask is supplied, this speficies the method for combining new mask with supplied mask: • "or" • "and" • "nor" • "nand" note: if mask is missing, "nand" results in negative of mask
bApply	when deriving a peakset/sample mask, a logical indicating that a new DBA object with the mask applied will be returned.

peakset	when deriving a peak/site mask, this specifies a peakset number, or a vector of peakset numbers. The resulting mask will indicate which of the overall sites were called as peaks in this peakset or set of peaksets. If a vector, the masks for each of the peaksets will be combined using the method specified in the combine parameter.
minValue	when deriving a peak/site mask, scores greater than this value will be considered as indicating that the site corresponds to a called peakset.

Details

MODE: Derive a a mask of peaksets/samples:

dba.mask(DBA, attribute, value, combine, mask, merge, bApply)

MODE: Derive a mask of peaks/sites:

dba.mask(DBA, combine, mask, merge,bApply, peakset, minValue)

Value

either a logical mask, or new DBA object if bApply is TRUE.

Note

dba automatically generates masks for each unique value of DBA_TISSUE, DBA_FACTOR, DBA_CONDITION, DBA_TREATMENT, DBA_CALLER, and DBA_REPLICATE. These are accessible using masks field of the DBA object (DBA$masks), and can be viewed using names(DBA$masks).

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)

# Pre-made masks
names(tamoxifen$masks)
dba.show(tamoxifen,tamoxifen$masks$MCF7)

# New masks
mcf7Mask = dba.mask(tamoxifen,DBA_TISSUE, "MCF7")
mcf7DerivedMask = dba.mask(tamoxifen,DBA_TISSUE,"TAMR",mask=mcf7Mask)
mcf7Derived = dba(tamoxifen,mcf7DerivedMask)
mcf7Derived
```

---

dba.overlap *Compute binding site overlaps (occupancy analysis)*

Description

Computes binding overlaps and co-occupancy statistics

Usage

```
dba.overlap(DBA, mask, mode=DBA_OLAP_PEAKS, minVal=0,
            contrast, method=DBA$config$AnalysisMethod, th=.1, bUsePval=FALSE,
            report, byAttribute, bCorOnly=TRUE, CorMethod="pearson",
            DataType=DBA$config$DataType)
```

Arguments

DBA	DBA object
mask	mask or vector of peakset numbers indicating a subset of peaksets to use (see dba.mask). When generating overlapping/unique peaksets, either two or three peaksets must be specified. If the mode type is DBA_OLAP_ALL, and a contrast is specified, a value of TRUE (mask=TRUE) indicates that all samples should be included (otherwise only those present in one of the contrast groups will be included).
mode	indicates which results should be returned (see MODES below). One of: • DBA_OLAP_PEAKS • DBA_OLAP_ALL • DBA_OLAP_RATE
minVal	minimum score value to be considered a "called" peak.
contrast	contrast number to use. Only specified if contrast data is to be used when mode=DBA_OLAP_ALL. See dba.show(DBA, bContrast=T) to get contrast numbers.
method	if contrast is specified and mode=DBA_OLAP_ALL, use data from method used for analysis: • DBA_EDGER • DBA_DESEQ • DBA_EDGER_BLOCK • DBA_DESEQ_BLOCK
th	if contrast is specified and mode=DBA_OLAP_ALL, significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be included. A value of 1 will include all binding sites, but only the samples included in the contrast.
bUsePval	if contrast is specified and mode=DBA_OLAP_ALL, logical indicating whether to use FDR (FALSE) or p-value (TRUE) for thresholding.
report	if contrast is specified and mode=DBA_OLAP_ALL, a report (obtained from dba.report) specifying the data to be used. If counts are included in the report (and a contrast is specified), the count data from the report will be used to compute correlations, rather than the scores in the global binding affinity matrix. If report is present, the method, th, and bUsePval parameters are ignored.
byAttribute	when computing co-occupancy statistics (DBA_OLAP_ALL), limit comparisons to peaksets with the same value for a specific attribute, one of: • DBA_ID • DBA_TISSUE • DBA_FACTOR • DBA_CONDITION • DBA_TREATMENT

- DBA_REPLICATE
- DBA_CONSENSUS
- DBA_CALLER
- DBA_CONTROL

`bCorOnly` when computing co-occupancy statistics (DBA_OLAP_ALL), logical indicating that only correlations, and not overlaps, should be computed. This is much faster if only correlations are desired (e.g. to plot the correlations using dba.plotHeatmap).

`CorMethod` when computing co-occupancy statistics (DBA_OLAP_ALL), method to use when computing correlations.

`DataType` if mode==DBA_OLAP_PEAKS, the class of object that peaksets should be returned as:

- DBA_DATA_GRANGES
- DBA_DATA_RANGEDDATA
- DBA_DATA_FRAME

Can be set as default behavior by setting DBA$config$DataType.

Details

MODE: Generate overlapping/unique peaksets:

dba.overlap(DBA, mask, mode=DBA_OLAP_PEAKS, minVal)

MODE: Compute correlation and co-occupancy statistics (e.g. for dba.plotHeatmap):

dba.overlap(DBA, mask, mode=DBA_OLAP_ALL, byAttribute, minVal, attributes, bCorOnly, CorMethod)

MODE: Compute correlation and co-occupancy statistics using significantly differentially bound sites (e.g. for dba.plotHeatmap):

dba.overlap(DBA, mask, mode=DBA_OLAP_ALL, byAttribute, minVal, contrast, method, th=, bUsePval, attributes, bCorOnly, CorMethod)

Note that the scores from the global binding affinity matrix will be used for correlations unless a report containing count data is specified.

MODE: Compute overlap rates at different stringency thresholds:

dba.overlap(DBA, mask, mode=DBA_OLAP_RATE, minVal)

Value

Value depends on the mode specified in the mode parameter.

If mode = DBA_OLAP_PEAKS, Value is an overlap record: a list of three peaksets for an A-B overlap, and seven peaksets for a A-B-C overlap:

`inAll` peaks in all peaksets

`onlyA` peaks unique to peakset A

`onlyB` peaks unique to peakset B

`onlyC` peaks unique to peakset C

`notA` peaks in both peaksets B and C but not peakset A

`notB` peaks in both peaksets A and C but not peakset B

`notC` peaks in both peaksets A and B but not peakset C

If mode = DBA_OLAP_ALL, Value is a correlation record: a matrix with a row for each pair of peaksets and the following columns:

A	peakset number of first peakset in overlap
B	peakset number of second peakset in overlap
onlyA	number of sites unique to peakset A
onlyB	number of sites unique to peakset B
inAll	number of peaks in both peakset A and B (merged)
R2	correlation value A vs B
Overlap	percentage overlap (number of overlapping sites divided by number of peaks unique to smaller peakset

If mode = DBA_OLAP_RATE, Value is a vector whose length is the number of peaksets, containing the number of overlapping peaks at the corresponding minOverlaps threshold (i.e., Value[1] is the total number of unique sites, Value[2] is the number of unique sites appearing in at least two peaksets, Value[3] the number of sites overlapping in at least three peaksets, etc.).

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
# default mode: DBA_OLAP_PEAKS -- get overlapping/non overlapping peaksets
mcf7 = dba.overlap(tamoxifen,tamoxifen$masks$MCF7&tamoxifen$masks$Responsive)
names(mcf7)
mcf7$inAll

# mode:  DBA_OLAP_ALL -- get correlation record
mcf7 = dba(tamoxifen,tamoxifen$masks$MCF7)
mcf7.corRec = dba.overlap(mcf7,mode=DBA_OLAP_ALL,bCorOnly=FALSE)
mcf7.corRec

# mode: DBA_OLAP_RATE -- get overlap rate vector
data(tamoxifen_peaks)
rate = dba.overlap(tamoxifen, mode=DBA_OLAP_RATE)
rate
plot(rate,type='b',xlab="# peaksets",ylab="# common peaks",
     main="Tamoxifen dataset overlap rate")
```

---

dba.peakset *Add a peakset to, or retrieve a peakset from, a DBA object*

Description

Adds a peakset to, or retrieves a peakset from, a DBA object

Usage

```
dba.peakset(DBA=NULL, peaks, sampID, tissue, factor, condition, treatment, replicate,
            control, peak.caller, reads=0, consensus=FALSE,
            bamReads, bamControl,
            normCol=4, bRemoveM=TRUE, bRemoveRandom=TRUE,
            minOverlap=2, bMerge=TRUE,
            bRetrieve=FALSE, writeFile, numCols=4,
            DataType=DBA$config$DataType)
```

Arguments

DBA	DBA object. Required unless creating a new DBA object by adding an initial peakset.
peaks	When adding a specified peakset: set of peaks, either a GRanges or RangedData object, or a peak dataframe or matrix (chr,start,end,score), or a filename where the peaks are stored. When adding a consensus peakset: a sample mask or vector of peakset numbers. If missing or NULL, a consensus is derived from all peaksets present in the model. See dba.mask, or dba.show to get peakset numbers. When adding all the peaks from one DBA object to another: a DBA object. In this case, the only other parameter to have an effect is minOverlap. When retrieving and/or writing a peakset: either a GRanges or RangedData object, or a peak dataframe or matrix (chr,start,end,score), or a peakset number; if NULL, retrieves/writes the full binding matrix.
sampID	ID string for the peakset being added; if missing, one is assigned (a serial number for a new peakset, or a concatenation of IDs for a consensus peakset).
tissue	tissue name for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of tissues).
factor	factor name for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of factors).
condition	condition name for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of conditions).
treatment	treatment name for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of treatment).
replicate	replicate number for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of replicate numbers).
control	control name for the peakset being added; if missing, one is assigned for a consensus peakset (a concatenation of control names).
peak.caller	peak caller name string. If peaks is specified as a file, this will control how it is interpreted. Supported values: • "raw": text file file; peak score is in fourth column • "bed": .bed file; peak score is in fifth column • "macs": MACS .xls file • "swembl": SWEMBL .peaks file • "bayes": bayesPeak file • "peakset": peakset written out using pv.writepeakset • "fp4": FindPeaks v4 if missing, a name is assigned for a consensus peakset (a concatenation of peak caller names).
reads	total number of ChIPed library reads for the peakset being added.
consensus	TRUE if peakset being added is made from overlap of other peaksets (set automatically when adding a consensus peakset).
bamReads	file path of the BAM/SAM/BED file containing the aligned reads for the peakset being added.
bamControl	file path of the BAM/SAM/BED file containing the aligned reads for the control used for the peakset being added.

normCol	peak column to normalize to 0...1 scale when adding a peakset; 0 indicates no normalization
bRemoveM	logical indicating whether to remove peaks on chrM when adding a peakset
bRemoveRandom	logical indicating whether to remove peaks on chrN_random when adding a peakset
minOverlap	the minimum number of peaksets a peak must be in to be included when adding a consensus peakset. When retrieving, if the peaks parameter is a vector (logical mask or vector of peakset numbers), a binding matrix will be retrieved including all peaks in at least this many peaksets.
bMerge	logical indicating whether global binding matrix should be compiled after adding the peakset. When adding several peaksets via successive calls to dba.peakset, it may be more efficient to set this parameter to FALSE and call dba(DBA) after all the peaksets have been added.
bRetrieve	logical indicating that a peakset is being retrieved and/or written, not added.
writeFile	file to write retrieved peakset.
numCols	number of columns to include when writing out peakset. First four columns are chr, start, end, score; the remainder are maintained from the original peakset. Ignored when writing out complete binding matrix.
DataType	The class of object for returned peaksets: • DBA_DATA_GRANGES • DBA_DATA_RANGEDDATA • DBA_DATA_FRAME Can be set as default behavior by setting DBA$config$DataType.

Details

MODE: Add a specified peakset:

dba.peakset(DBA=NULL, peaks, sampID, tissue, factor, condition, replicate, control, peak.caller, reads, consensus, bamReads, bamControl, normCol, bRemoveM, bRemoveRandom)

MODE: Add a consensus peakset (derived from overlapping peaks in peaksets already present):

dba.peakset(DBA, peaks, minOverlap)

MODE: Retrieve a peakset:

dba.peakset(DBA, peaks, bRetrieve=T)

MODE: Write a peakset out to a file:

dba.peakset(DBA, peaks, bRetrieve=T, writeFile, numCols)

Value

DBA object when adding a peakset. Peakset matrix or RangedData object when retrieving and/or writing a peakset.

Author(s)

Rory Stark

Examples

```
# create a new DBA object by adding three peaksets
mcf7 = dba.peakset(NULL,
                   peaks=system.file("extra/peaks/MCF7_ER_1.bed.gz", package="DiffBind"),
                   sampID="MCF7.1",tissue="MCF7",factor="ER",condition="Responsive",replicate=1)
mcf7 = dba.peakset(mcf7,
                   peaks=system.file("extra/peaks/MCF7_ER_2.bed.gz", package="DiffBind"),
                   sampID="MCF7.2",tissue="MCF7",factor="ER",condition="Responsive",replicate=2)
mcf7 = dba.peakset(mcf7,
                   peaks=system.file("extra/peaks/MCF7_ER_3.bed.gz", package="DiffBind"),
                   sampID="MCF7.3",tissue="MCF7",factor="ER",condition="Responsive",replicate=3)
mcf7

#retrieve peaks that are in all three peaksets
mcf7.consensus = dba.peakset(mcf7, 1:3, minOverlap=3, bRetrieve=TRUE)
mcf7.consensus

#add a consensus peakset -- peaks in all three replicates
mcf7 = dba.peakset(mcf7, 1:3, minOverlap=3,sampID="MCF7_3of3")
mcf7

#retrieve the consensus peakset as RangedData object
mcf7.consensus = dba.peakset(mcf7,mcf7$masks$Consensus,bRetrieve=TRUE)
mcf7.consensus
```

`dba.plotBox` *Boxplots*

Description

Boxplots for read count distributions within differentially bound sites

Usage

```
dba.plotBox(DBA, contrast=1, method=DBA$config$AnalysisMethod,
            th=0.1, bUsePval=FALSE, bNormalized=TRUE,
            attribute=DBA_GROUP,
            bAll=FALSE, bAllIncreased=FALSE, bAllDecreased=FALSE,
            bDB=TRUE, bDBIncreased=TRUE, bDBDecreased=TRUE,
            pvalMethod=wilcox.test,  bReversePos=FALSE, attribOrder,
            vColors, varwidth=TRUE, notch=TRUE, ...)
```

Arguments

`DBA`	DBA object.
`contrast`	number of contrast to use for boxplot.
`method`	method used for analysis (used in conjunction with contrast): • DBA_EDGER • DBA_DESEQ

- DBA_EDGER_BLOCK
- DBA_DESEQ_BLOCK

`th` significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be included in the boxplot.

`bUsePval` logical indicating whether to use FDR (FALSE) or p-value (TRUE) for thresholding.

`bNormalized` logical indicating that normalized data (using normalization factors computed by differential analysis method) should be plotted. FALSE uses raw count data.

`attribute` attribute to use for determining groups of samples. Default (DBA_GROUP) plots the two groups used in the contrast. Possible values:

- DBA_GROUP
- DBA_ID
- DBA_TISSUE
- DBA_FACTOR
- DBA_CONDITION
- DBA_TREATMENT
- DBA_REPLICATE
- DBA_CONSENSUS
- DBA_CALLER
- DBA_CONTROL

`bAll` logical indicating if plot should include a set of boxplots using all counts, regardless of whether or not they pass the significance threshold.

`bAllIncreased` logical indicating if plot should include a set of boxplots using all counts that increase in affinity, regardless of whether or not they pass the significance threshold.

`bAllDecreased` logical indicating if plot should include a set of boxplots using all counts that decrease in affinity, regardless of whether or not they pass the significance threshold.

`bDB` logical indicating if plot should include a set of boxplots using all counts in significantly differentially bound sites (i.e. those that pass the significance threshold), regardless of whether they increase or decrease in affinity.

`bDBIncreased` logical indicating if plot should include a set of boxplots using all counts in significantly differentially bound sites that increase in affinity.

`bDBDecreased` logical indicating if plot should include a set of boxplots using all counts in significantly differentially bound sites that decrease in affinity.

`pvalMethod` method to use when computing matrix of p-values. If NULL, no matrix is computed, and NULL is returned; this may speed up processing if there are many boxplots.

`bReversePos` logical indicating if the default definition of positive affinity (higher affinity in the second group of the contrast) should be reversed (i.e. positive affinity is defined as being higher in the first group of the contrast).

`attribOrder` vector of group numbers used to change the order that groups are plotted. If NULL, default order is used (group order for DBA_GROUP, and the order the attribute values appear for other values of attribute).

`vColors` vector of custom colors; if absent, default colors will be used.

`varwidth` passed to boxplot

`notch` passed to boxplot

`...` other arguments passed to boxplot

Details

Draws a boxplot showing distributions of read counts for various groups of samples under various conditions. In default mode, draws six boxes: one pair of boxes showing the distribution of read counts within all significantly differentially bound sites (one box for each sample group), one pair of boxes showing the distribution of read counts for significantly differentially bound sites that increase affinity in the second group, and a second pair of boxes showing the distribution of read counts for significantly differentially bound sites that have higher mean affinity in the first group.

Value

if pvalMethod is not NULL, returns a matrix of p-values indicating the significance of the difference between each pair of distributions.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_analysis)

#default boxplot includes all DB sites, then divided into those increasing
# affinity in each group
dba.plotBox(tamoxifen)

# plot non-normalized data for DB sites by tissue
# (changing order to place Resistant samples last)
dba.plotBox(tamoxifen, attribute=DBA_CONDITION, bDBIncreased=FALSE,
            bDBDecreased=FALSE, attribOrder=c(2,1), bNormalized=FALSE)
```

---

dba.plotHeatmap *Draw a binding site heatmap*

Description

Draws a binding site heatmap

Usage

```
dba.plotHeatmap(DBA, attributes=DBA$attributes, maxSites=1000, minval, maxval,
                contrast, method=DBA$config$AnalysisMethod,
                th=.1, bUsePval=FALSE, report, score,
                mask, sites, sortFun,
                correlations=TRUE, olPlot=DBA_COR,
                margin=10, colScheme="Greens", distMethod="pearson",
                ...)
```

Arguments

DBA	DBA object.
attributes	attribute or vector of attributes to use for column labels: • DBA_ID • DBA_TISSUE • DBA_FACTOR • DBA_CONDITION • DBA_TREATMENT • DBA_REPLICATE • DBA_CONSENSUS • DBA_CALLER • DBA_CONTROL
maxSites	maximum number of binding sites to use in heatmap. Only used when not drawing a correlation heatmap (correlations=FALSE)
minval	Set all scores less than this to minval
maxval	Set all scores greater than this to maxval
contrast	number of contrast to report on; if present, draws a heatmap based on a differential binding affinity analysis (see dba.analyze). See dba.show(DBA, bContrast=T) to get contrast numbers.
method	analysis method (used in conjunction with contrast): • DBA_EDGER • DBA_DESEQ • DBA_EDGER_BLOCK • DBA_DESEQ_BLOCK
th	significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be included in the report (subject to maxSites). Used in conjunction with contrast.
bUsePval	logical indicating whether to use FDR (FALSE) or p-value (TRUE) for thresholding. Used in conjunction with contrast.
report	report (obtained from dba.report) specifying the data to be used . If this is present, the method, th, and bUsePval parameters are ignored. Used in conjunction with contrast.
score	Score to use for count data. Only used when plotting the global binding matrix (no contrast specified). One of: • DBA_SCORE_READS • DBA_SCORE_READS_MINUS • DBA_SCORE_READS_FOLD • DBA_SCORE_RPKM • DBA_SCORE_RPKM_FOLD • DBA_SCORE_TMM_READS_FULL • DBA_SCORE_TMM_READS_EFFECTIVE • DBA_SCORE_TMM_MINUS_FULL • DBA_SCORE_TMM_MINUS_EFFECTIVE
mask	mask indicating a subset of peaksets to use when using global binding matrix (contrast is missing). See dba.mask.

sites	logical vector indicating which sites to include; first maxSites of these. Only relevant when using global binding matrix (contrast is missing).
sortFun	function taking a vector of scores and returning a single value. Only relevant when using global binding matrix (contrast is missing). If present, the global binding matrix will be sorted (descending) on the results, and the first maxSites used in the heatmap. Recommended sort function options include sd, mean, median, min.
correlations	logical indicating that a correlation heatmap should be plotted (TRUE). If FALSE, a binding heatmap of scores/reads is plotted. This parameter can also be set to a correlation record; see dba.overlap(mode=DBA_OLAP_ALL), in which case a correlation heatmap is plotted based on the specified correlation record, using the statistic specified in olPlot.
olPlot	if correlations is specified as a dataframe returned by dba.overlap, indicates which statistic to plot. One of: • DBA_COR Correlation • DBA_OLAP Percentage overlap • DBA_INALL number of peaks common to both samples
margin	margin size of plot
colScheme	Color scheme; see colorRampPalette RColorBrewer
distMethod	distance method for clustering; see Dist amap.
...	passed on to heatmap.2 (gplots), e.g. scale etc.

Details

MODE: Correlation Heatmap plot using statistics for global binding matrix:

dba.plotHeatmap(DBA, attributes=DBA$attributes, minval, maxval, correlations, olPlot, colScheme="Greens", distMethod="pearson", ...)

MODE: Correlation Heatmap plot using statistics for significantly differentially bound sites:

dba.plotHeatmap(DBA, attributes=DBA$attributes, minval, maxval, contrast, method=DBA_EDGER, th=.1, bUsePval=F, overlaps, olPlot=DBA_COR, colScheme="Greens", distMethod="pearson", ...)

MODE: Binding heatmap plot using significantly differentially bound sites:

dba.plotHeatmap(DBA, attributes, maxSites, minval, maxval, contrast, method, th, bUsePval, correlations=FALSE, colScheme, distMethod, ...)

MODE: Binding heatmap plot using the global binding matrix:

dba.plotHeatmap(DBA, attributes, maxSites, minval, maxval, mask, sites, correlations=FALSE, sortFun, colScheme, distMethod, ...)

Value

if correlations is not FALSE, the overlap/correlation matrix is returned.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
# peak overlap correlation heatmap
dba.plotHeatmap(tamoxifen)

data(tamoxifen_counts)
# counts correlation heatmap
dba.plotHeatmap(tamoxifen)

data(tamoxifen_analysis)
#correlation heatmap based on all normalized data
dba.plotHeatmap(tamoxifen,contrast=1,th=1)

#correlation heatmap based on DB sites only
dba.plotHeatmap(tamoxifen,contrast=1)

#binding heatmap based on DB sites
dba.plotHeatmap(tamoxifen,contrast=1,correlations=FALSE)

#binding heatmap based on 1,000 sites with highest variance
dba.plotHeatmap(tamoxifen,contrast=1,th=1,correlations=FALSE,sortFun=var)
```

---

dba.plotMA *Generate MA and scatter plots of differential binding analysis results*

Description

Generates MA and scatter plots of differential binding analysis results.

Usage

```
dba.plotMA(DBA, contrast=1, method=DBA$config$AnalysisMethod,
           th=.1, bUsePval=FALSE,
           bNormalized=TRUE, factor="", bXY=FALSE, dotSize=.33, ...)
```

Arguments

DBA	DBA object, on which dba.analyze should have been successfully run.
contrast	number of contrast to report on. See dba.show(DBA, bContrast=T) to get contrast numbers.
method	method or vector of methods to plot results for: • DBA_EDGER • DBA_DESEQ • DBA_EDGER_BLOCK
th	significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be colored red in the plot
bUsePval	logical indicating whether to use FDR (FALSE) or p-value (TRUE) for thresholding.
bNormalized	logical indicating whether to plot normalized data using normalization factors computed by differential analysis method (TRUE) or raw read counts (FALSE).

factor	string to be prepended to plot main title; e.g. factor name.
bXY	logical indicating whether to draw MA plot (FALSE) or XY scatter plot (TRUE).
dotSize	size of points on plot (cex).
...	passed to plot.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_analysis)

# default MA plot
dba.plotMA(tamoxifen)

#XY plots (with raw and normalized data)
par(mfrow=c(1,2))
dba.plotMA(tamoxifen,bXY=TRUE,bNormalized=FALSE)
dba.plotMA(tamoxifen,bXY=TRUE,bNormalized=TRUE)
```

---

dba.plotPCA *PCA plot*

Description

Principal Component Analysis plot

Usage

```
dba.plotPCA(DBA, attributes, minval, maxval,
            contrast, method=DBA$config$AnalysisMethod,
            th=.1, bUsePval=FALSE, report, score,
            mask, sites, cor=FALSE,
            b3D=FALSE, vColors, dotSize, ...)
```

Arguments

DBA	DBA object.
attributes	attribute or vector of attributes to use to color plotted points. Each unique combination of attribute values will be assigned a color. Chosen from: • DBA_GROUP • DBA_ID • DBA_TISSUE • DBA_FACTOR • DBA_CONDITION • DBA_TREATMENT • DBA_REPLICATE • DBA_CONSENSUS • DBA_CALLER

- DBA_CONTROL

Note that DBA_GROUP is a special attribute which will result in samples from each group in a contrast being colored separately.

`minval` Set all scores less than this to minval

`maxval` Set all scores greater than this to maxval

`contrast` number of contrast to use for PCA; if present, plots a PCA based on a differential binding affinity analysis (see dba.analyze). See dba.show(DBA, bContrast=T) to get contrast numbers. If missing, uses scores in the main binding matrix.

`method` method used for analysis (used in conjunction with contrast):

- DBA_EDGER
- DBA_DESEQ
- DBA_EDGER_BLOCK
- DBA_DESEQ_BLOCK

`th` significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be included in the PCA, subject to maxVal. Used in conjunction with contrast.

`bUsePval` if TRUE, uses p-value instead of FDR for thresholding. Used in conjunction with contrast.

`report` report (obtained from dba.report) specifying the data to be used . If this is present, the method, th, and bUsePval parameters are ignored.

`score` Score to use for count data. Only used when plotting the global binding matrix (no contrast specified). One of:

- DBA_SCORE_READS
- DBA_SCORE_READS_MINUS
- DBA_SCORE_READS_FOLD
- DBA_SCORE_RPKM
- DBA_SCORE_RPKM_FOLD
- DBA_SCORE_TMM_READS_FULL
- DBA_SCORE_TMM_READS_EFFECTIVE
- DBA_SCORE_TMM_MINUS_FULL
- DBA_SCORE_TMM_MINUS_EFFECTIVE

`mask` mask indicating a subset of peaksets to use when using global binding matrix (contrast is missing). See dba.mask.

`sites` logical vector indicating which sites to include in PCA. Only relevant when using global binding matrix (contrast is missing).

`cor` a logical value indicating whether the calculation should use the correlation matrix or the covariance matrix. Passed into princomp.

`b3D` logical indicating that three principal components should be plotted (requires package{rgl}). If FALSE, the first two principal components are plotted.

`vColors` vector of custom colors; is absent, default colors will be used.

`dotSize` size of dots to plot; is absent, a default will be calculated.

`...` arguments passed to plot or plot3d (rgl).

Details

MODE: PCA plot using significantly differentially bound sites:

dba.plotPCA(DBA, attributes, minval, maxval, contrast, method, th, bUsePval, b3D=F, vColors, dotSize, ...)

MODE: PCA plot using global binding matrix:

dba.plotPCA(DBA, attributes, minval, maxval, mask, sites, b3D=F, vColors, dotSize, ...)

Value

matrix with color legend

Note

uses rgl package for 3D plots (if available)

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)

# peakcaller scores PCA
dba.plotPCA(tamoxifen)

# raw count correlation PCA
data(tamoxifen_analysis)
dba.plotPCA(tamoxifen)

#PCA based on normalized data for all sites
dba.plotPCA(tamoxifen,contrast=1,th=1)

#PCA based on DB sites only
par(mfrow=c(1,2))
dba.plotPCA(tamoxifen,contrast=1)
dba.plotPCA(tamoxifen,contrast=1,attributes=DBA_TISSUE)
```

---

dba.plotVenn *Draw 2-way or 3-way Venn diagrams of overlaps*

Description

Draws 2-way or 3-way Venn diagrams of overlaps

Usage

```
dba.plotVenn(DBA, mask, overlaps, label1, label2, label3, ...)
```

Arguments

DBA	DBA object; if present, only the mask parameter will apply.
mask	mask or vector of peakset numbers indicating which peaksets to include in Venn diagram. Only 2 or 3 peaksets should be included. See dba.mask. Only one of mask or overlaps is used.
overlaps	overlap record, as computed by dba.overlap(Report=DBA_OLAP_PEAKS). Only one of mask or overlaps is used.
label1	label for first peakset in diagram
label2	label for second peakset in diagram
label3	label for third peakset in diagram
...	arguments passed on to vennDiagram{limma}

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)

par(mfrow=c(2,2))
# 2-way Venn
dba.plotVenn(tamoxifen,6:7)
dba.plotVenn(tamoxifen,tamoxifen$masks$ZR75)

# 3-way Venn (done two different ways)
dba.plotVenn(tamoxifen,tamoxifen$masks$MCF7&tamoxifen$masks$Responsive)
olaps = dba.overlap(tamoxifen,tamoxifen$masks$MCF7&tamoxifen$masks$Responsive)
dba.plotVenn(tamoxifen,overlaps=olaps,
             label1="Rep 1",label2="Rep 2",label3="Rep 3",main="MCF7 (Responsive) Replicates")

#Venn of overlaps
Responsive=dba(tamoxifen,tamoxifen$masks$Responsive)
Responsive
Responsive = dba.peakset(Responsive,1:3,sampID="MCF7")
Responsive = dba.peakset(Responsive,4:5,sampID="T47D")
Responsive = dba.peakset(Responsive,6:7,sampID="ZR75")
dba.plotVenn(Responsive,Responsive$masks$Consensus)
```

---

dba.report *Generate a report for a differential binding affinity analysis*

Description

Generates a report for a differential binding affinity analysis

Usage

```
dba.report(DBA, contrast=1, method=DBA$config$AnalysisMethod,
           th=.1, bUsePval=FALSE, fold=0,  bNormalized=TRUE,
           bCalled=FALSE, bCounts=FALSE, bCalledDetail=FALSE,
           file,initString=DBA$config$reportInit,ext='csv',
           DataType=DBA$config$DataType)
```

Arguments

DBA	DBA object. A differential binding affinity analysis needs to have been previously carried out (see dba.analyze).
contrast	contrast number to report on. See dba.show(DBA, bContrast=T) to get contrast numbers.
method	method used for analysis: • DBA_EDGER • DBA_DESEQ • DBA_EDGER_BLOCK
th	significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be included in the report. A value of 1 will include all binding sites in the report.
bUsePval	logical indicating whether to use FDR (FALSE) or p-value (TRUE) for thresholding.
fold	only sites with an absolute Fold value greater than equal to this will be included in the report.
bNormalized	logical indicating that normalized data (using normalization factors computed by differential analysis method) should be reported. FALSE uses raw count data.
bCalled	logical indicating that peak caller status should be included (if available from a previous call to dba.count(bCalledMasks=TRUE)). This will add a column for each group, each indicating the number of samples in the group identified as a peak in the original peaksets.
bCounts	logical indicating that count data for individual samples should be reported as well as group statistics. Columns are added for each sample in the first group, followed by columns for each sample in the second group.
bCalledDetail	logical indicating that peak caller status should be included for each sample (if available). Columns are added for each sample in the first group, followed by columns for each sample in the second group.
file	if present, also save the report to a comma separated value (csv) file, using this filename.
initString	if saving to a file, pre-pend this string to the filename.
ext	if saving to a file, append this extension to the filename.
DataType	The class of object for returned report: • DBA_DATA_GRANGES • DBA_DATA_RANGEDDATA • DBA_DATA_FRAME Can be set as default behavior by setting DBA$config$DataType.

Value

A report dataframe or RangedData object, with a row for each binding site within the thresholding parameters, and the following columns:

Chr	Chromosome of binding site
Start	Starting base position of binding site
End	End base position of binding site
Conc	Concentration – mean (log) reads across all samples in both groups
Conc_group1	Group 1 Concentration – mean (log) reads across all samples first group
Conc_group2	Group 2 Concentration – mean (log) reads across all samples in second group
Fold	Fold difference – mean fold difference of binding affinity of group 1 over group 2 (Conc1 - Conc2). Absolute value indicates magnitude of the difference, and sign indicates which one is bound with higher affinity, with a positive value indicating higher affinity in the first group
p-value	p-value calculation – statistic indicating significance of difference (likelihood difference is not attributable to chance)
FDR	adjusted p-value calculation – p-value subjected to multiple-testing correction

If bCalled is TRUE and caller status is available, two more columns will follow:

Called1	Number of samples in group 1 that identified this binding site as a peak
Called2	Number of samples in group 2 that identified this binding site as a peak

If bCounts is TRUE, a column will be present for each sample in group 1, followed by each sample in group 2. The Sample ID will be used as the column header. This column contains the read counts for the sample.

If bCalledDetail is TRUE, a column will be present for each sample in group 1, followed by each sample in group 2. The Sample ID will be used as the column header. This column contains a "+" to indicate for which sites the sample was called as a peak, and a "-" if it was not so identified.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_analysis)

#Retrieve DB sites with FDR <  0.1
tamoxifen.DB = dba.report(tamoxifen)
tamoxifen.DB

#Retrieve DB sites with p-value <  0.05 and Fold > 2
tamoxifen.DB = dba.report(tamoxifen,th=.05,bUsePval=TRUE,fold=2)
tamoxifen.DB

#Retrieve all sites with confidence stats
# and how many times each site was identified as a peak
tamoxifen.DB = dba.report(tamoxifen, th=1, bCalled=TRUE)
tamoxifen.DB

#Retrieve all sites with confidence stats and normalized counts
```

```
tamoxifen.DB = dba.report(tamoxifen,th=1,bCounts=TRUE)
tamoxifen.DB

#Retrieve all sites with confidence stats and raw counts
tamoxifen.DB = dba.report(tamoxifen,th=1,bCounts=TRUE,bNormalized=FALSE)
tamoxifen.DB
```

---

dba.save *save DBA object*

Description

Writes out DBA object

Usage

```
dba.save(DBA, file='DBA', dir='.', pre='dba_', ext='RData', bMinimize=FALSE)
```

Arguments

DBA	DBA object
file	main filename
dir	directory to save model in
pre	string to pre-pend to filename
ext	extensions to use
bMinimize	logical indicating saved DBA object should be compressed as much as possible.

Value

string containing full path and filename.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
savefile = dba.save(tamoxifen,'tamoxifenPeaks')
savefile
tamoxifen = dba.load('tamoxifenPeaks')
unlink(savefile)
```

dba.show *List attributes of peaksets of contrasts associated with a DBA object*

Description

Returns attributes of peaksets and/or contrasts associated with a DBA object.

Usage

```
dba.show(DBA, mask, attributes, bContrasts=FALSE, th=0.1, bUsePval=FALSE)
```

Arguments

DBA	DBA object
mask	mask of peaksets for which to get attributes (used when obtaining peakset attributes, i.e. bContrasts=FALSE).
attributes	attribute or vector of attributes to retrieve. Number of intervals is always shown. Used when obtaining peakset attributes, i.e. bContrasts=FALSE. Values: • DBA_ID • DBA_TISSUE • DBA_FACTOR • DBA_CONDITION • DBA_CONDITION • DBA_REPLICATE • DBA_CONSENSUS • DBA_CALLER • DBA_CONTROL
bContrasts	logical indicating whether peaksets or contrast attributes are to be retrieved. TRUE retrieves a dataframe of contrast information instead of peakset attributes. If no contrasts are set, returns possible contrasts. See dba.contrast.
th	if bContrasts is TRUE, then th is used as the threshold for determining how many significant sites there are for each contrast. Only relevant when obtaining contrast attributes (bContrasts=TRUE) and dba.analyze has been run.
bUsePval	logical indicating that p-values will be used (along with th) to determine how many significant sites there are for each contrast; if FALSE, adjusted p-values (FDR) are used. Only relevant when obtaining contrast attributes (bContrasts=TRUE) and dba.analyze has been run.

Details

MODE: Return attributes of peaksets associated with a DBA object:

dba.show(DBA, mask, attributes)

MODE: Return contrasts associated with a DBA object:

dba.show(DBA,bContrasts=T, th, bUsePval)

Value

dataframe with peakset attributes.

If bContrasts == FALSE, each row represents a peakset, and each column is an attributes, with the final column, Intervals, indicating how many sites there are in the peakset.

If bContrasts == TRUE, each row represent a contrast, with the following columns:

`Group1`	Label for first group of contrast
`Members1`	Number of samples in first group of contrast
`Group2`	Label for first group of contrast
`Members3`	Number of samples in first group of contrast

if dba.analyze has been successfully run, there there will be up to two more columns showing the number of significant differentially bound (DB) sites identified for

`DB.edgeR`	Number of significantly differentially bound sites identified using edgeR
`DB.DESeq`	Number of significantly differentially bound sites identified using DESeq

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
dba.show(tamoxifen)
dba.show(tamoxifen,tamoxifen$masks$Responsive)
dba.show(tamoxifen,attributes=c(DBA_TISSUE,DBA_REPLICATE,DBA_CONDITION))

data(tamoxifen_counts)
tamoxifen = dba.contrast(tamoxifen)
dba.show(tamoxifen,bContrasts=TRUE)

#alternatively:
tamoxifen
```

`DiffBind -- DBA global constant variables`

Constant variables used in DiffBind package

Description

Constant variables used in DiffBind package

Usage

```
DBA_ID
DBA_FACTOR
DBA_TISSUE
DBA_CONDITION
DBA_TREATMENT
DBA_REPLICATE
```

```
DBA_CALLER
DBA_CONSENSUS
DBA_CONTROL

DBA_GROUP

DBA_OLAP_PEAKS
DBA_OLAP_ALL
DBA_OLAP_RATE

DBA_SCORE_READS
DBA_SCORE_READS_MINUS
DBA_SCORE_READS_FOLD
DBA_SCORE_RPKM
DBA_SCORE_RPKM_FOLD
DBA_SCORE_TMM_READS_FULL
DBA_SCORE_TMM_READS_EFFECTIVE
DBA_SCORE_TMM_MINUS_FULL
DBA_SCORE_TMM_MINUS_EFFECTIVE

DBA_EDGER
DBA_DESEQ
DBA_EDGER_BLOCK
DBA_DESEQ_BLOCK
DBA_EDGER_CLASSIC
DBA_DESEQ_CLASSIC
DBA_EDGER_GLM
DBA_DESEQ_GLM

DBA_DATA_FRAME
DBA_DATA_GRANGES
DBA_DATA_RANGEDDATA
```

Arguments

DBA_ID	DBA peakset metadata: Peakset ID
DBA_FACTOR	DBA peakset metadata: Factor
DBA_TISSUE	DBA peakset metadata: Tissue
DBA_CONDITION	DBA peakset metadata: Condition
DBA_TREATMENT	DBA peakset metadata: Treatment
DBA_REPLICATE	DBA peakset metadata: Replicate
DBA_CALLER	DBA peakset metadata: Peak Caller
DBA_CONSENSUS	DBA peakset metadata: Is this a consensus peakset?
DBA_CONTROL	DBA peakset metadata: ID of Control sample
DBA_GROUP	DBA peakset metadata: color PCA plot using contras groups
DBA_OLAP_PEAKS	dba.overlap mode: return overlapping/unique peaksets
DBA_OLAP_ALL	dba.overlap mode: return report of correlations/overlaps for each pair of samples
DBA_OLAP_RATE	dba.overlap mode: return overlap rates

DBA_SCORE_READS
: dba.count score is number of reads in ChIP

DBA_SCORE_READS_FOLD
: dba.count score is number of reads in ChIP divided by number of reads in Control

DBA_SCORE_READS_MINUS
: dba.count score is number of reads in ChIP minus number of reads in Control

DBA_SCORE_RPKM
: dba.count score is RPKM of ChIP

DBA_SCORE_RPKM_FOLD
: dba.count score is RPKM of ChIP divided by RPKM of Control

DBA_SCORE_TMM_READS_FULL
: dba.count score is TMM normalized (using edgeR), using ChIP read counts and Full Library size

DBA_SCORE_TMM_READS_EFFECTIVE
: dba.count score is TMM normalized (using edgeR), using ChIP read counts and Effective Library size

DBA_SCORE_TMM_MINUS_FULL
: dba.count score is TMM normalized (using edgeR), using ChIP read counts minus Control read counts and Full Library size

DBA_SCORE_TMM_MINUS_EFFECTIVE
: dba.count score is TMM normalized (using edgeR), using ChIP read counts minus Control read counts and Effective Library size

DBA_EDGER
: differential analysis method: edgeR (default: DBA_EDGER_GLM)

DBA_DESEQ
: differential analysis method: DESeq (default: DBA_DESEQ_CLASSIC)

DBA_EDGER_CLASSIC
: differential analysis method: "classic" edgeR for two-group comparisons

DBA_DESEQ_CLASSIC
: differential analysis method: "classic" DESeq for two-group comparisons

DBA_EDGER_GLM
: differential analysis method: use GLM in edgeR for two-group comparisons

DBA_DESEQ_GLM
: differential analysis method: use GLM in DESeq for two-group comparisons

DBA_EDGER_BLOCK
: differential analysis method: edgeR with blocking factors (GLM)

DBA_DESEQ_BLOCK
: differential analysis method: DESeq with blocking factors (GLM)

DBA_DATA_GRANGES
: Use GRanges class for peaksets and reports. This is the default (DBA$config$DataType = DBA_DATA_GRANGES).

DBA_DATA_RANGEDDATA
: Use RangedData class for peaksets and reports. Can be set as default (DBA$config$DataType = DBA_DATA_RANGEDDATA).

DBA_DATA_FRAME
: Use data.frame class for peaksets and reports. Can be set as default (DBA$config$DataType = DBA_DATA_FRAME).

Note

Variables with ALL CAP names are used as constants within DiffBind.

Author(s)

Rory Stark

Index