Package: DiffBind (via r-universe)

November 8, 2026

Type Package

Version 3.22.2

Title Differential Binding Analysis of ChIP-Seq Peak Data

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

License Artistic-2.0

LazyLoad yes

Depends R (>= 4.0), GenomicRanges, SummarizedExperiment

Imports RColorBrewer, amap, gplots, grDevices, limma, GenomicAlignments, locfit, stats, utils, IRanges, lattice, systemPipeR, tools, Rcpp, dplyr, ggplot2, BiocParallel, parallel, S4Vectors, Rsamtools (>= 2.13.1), DESeq2, methods, graphics, ggrepel, apeglm, ashr, GreyListChIP

Suggests BiocStyle, testthat, xtable, rgl, XLConnect, edgeR, csaw, BSgenome, GenomeInfoDb, rtracklayer, grid

LinkingTo Rhtslib (>= 1.99.1), Rcpp

SystemRequirements GNU make

Collate core.R parallel.R model.R counts.R contrast.R normalize.R analyze.R analyze_deseq2.R analyze_edgeR.R blacklist.R report.R plots.R plotProfile.R io.R helper.R utils.R RcppExports.R cpp_wrapper.R DBA.R

biocViews Sequencing, ChIPSeq,ATACSeq, DNaseSeq, MethylSeq, RIPSeq, DifferentialPeakCalling, DifferentialMethylation, GeneRegulation, HistoneModification, PeakDetection, BiomedicalInformatics, CellBiology, MultipleComparison, Normalization, ReportWriting, Epigenetics, FunctionalGenomics

URL https://bioconductor.org/packages/DiffBind/

Config/pak/sysreqs cmake make libbz2-dev libicu-dev libjpeg-dev liblzma-dev libpng-dev libuv1-dev libxml2-dev libssl-dev xz-utils zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-07-20 14:16:58 UTC

RemoteUrl https://github.com/bioc/DiffBind

RemoteRef RELEASE_3_23

RemoteSha 918d1a717762b633e1af165bbadec95194546a60

Contents

DiffBind-package . . . 2
dba . . . 3
DBA object methods . . . 8
DBA tamoxifen resistance dataset . . . 11
dba.analyze . . . 12
dba.blacklist . . . 14
dba.contrast . . . 17
dba.count . . . 22
dba.load . . . 26
dba.mask . . . 27
dba.normalize . . . 29
dba.overlap . . . 34
dba.peakset . . . 37
dba.plotBox . . . 42
dba.plotHeatmap . . . 44
dba.plotMA . . . 48
dba.plotPCA . . . 51
dba.plotProfile . . . 54
dba.plotVenn . . . 57
dba.plotVolcano . . . 61
dba.report . . . 62
dba.save . . . 66
dba.show . . . 68
DiffBind – DBA global constant variables . . . 70
DiffBind3 . . . 77

Index **79**

`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.blacklist:	Filter peaks using blacklists and greylists
dba.count:	Count reads in binding sites
dba.contrast:	Establish design and contrast(s) for analysis
dba.normalize:	Normalize count data for analysis
dba.analyze:	Execute quantitative analysis
dba.report:	Generate results 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.plotVolcano:	Volcano plot
dba.plotProfile:	Peak profile heatmaps
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 @at@ cruk.cam.ac.uk> and Gord Brown

---

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(AnalysisMethod=DBA_DESEQ2,th=0.05,
                      DataType=DBA_DATA_GRANGES, RunParallel=TRUE,
                      minQCth=15, fragmentSize=125,
                      bCorPlot=FALSE, reportInit="DBA",
                      bUsePval=FALSE, design=TRUE,
                      doBlacklist=TRUE, doGreylist=TRUE),
    peakCaller="raw", peakFormat, scoreCol, bLowerScoreBetter,
    filter, skipLines=0,
    bAddCallerConsensus=FALSE,
    bRemoveM=TRUE, bRemoveRandom=TRUE,
    bSummarizedExperiment=FALSE,
    attributes, dir)
```

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 vector 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. If minOverlap is between zero and one, peak will be included from at least this proportion of peaksets.
sampleSheet	data frame containing sample sheet, or file name of sample sheet to load (ignored if DBA is specified). Columns names in sample sheet may include:

- SampleID: Identifier string for sample. Must be unique for each sample.
- Tissue: Identifier string for tissue type
- Factor: Identifier string for factor
- Condition: Identifier string for condition
- Treatment: Identifier string for treatment
- 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
- Spikein: file path for bam file containing aligned spike-in reads
- ControlID: Identifier string for control sample
- 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
  - "narrow": default peak.format: narrowPeaks file
  - "macs": MACS .xls file
  - "swembl": SWEMBL .peaks file
  - "bayes": bayesPeak file
  - "peakset": peakset written out using pv.writepeakset
  - "fp4": FindPeaks v4
- `PeakFormat`: string indicating format for peak files; see PeakCaller and `dba.peakset`
- `ScoreCol`: column in peak files that contains peak scores
- `LowerBetter`: logical indicating that lower scores signify better peaks
- `Counts`: file path for externally computed read counts; see `dba.peakset` (`counts` parameter)

For sample sheets loaded from a file, the accepted formats are comma-separated values (column headers, followed by one line per sample), or Excel-formatted spreadsheets (`.xls` or `.xlsx` extension). Leading and trailing white space will be removed from all values, with a warning.

`config` `list` containing configuration options, or file name of config file to load when constructing a new DBA object from a sample sheet. `NULL` indicates no config file.

See DBA-config for full set of options. Relevant fields include:

- `AnalysisMethod`: either `DBA_DESEQ2` or `DBA_EDGER`.
- `th`: default threshold for reporting and plotting analysis results.
- `DataType`: default class for peaks and reports (`DBA_DATA_GRANGES`, `DBA_DATA_RANGEDDATA`, or `DBA_DATA_FRAME`).
- `RunParallel`: logical indicating if counting and analysis operations should be run in parallel using multicore by default.
- `minQCth`: numeric, for filtering reads based on mapping quality score; only reads with a mapping quality score greater than or equal to this will be counted.
- `fragmentSize`: numeric with mean fragment size. Reads will be extended to this length before counting overlaps. May be a vector of lengths, one for each sample.
- `bCorPlot`: logical indicating that a correlation heatmap should be plotted automatically
- `ReportInit`: string to append to the beginning of saved report file names.
- `bUsePval`: logical, default indicating whether to use FDR (`FALSE`) or p-values (`TRUE`).
- `doBlacklist`: logical, whether to attempt to find and apply a blacklist if none is present when running `dba.analyze`.
- `doGreylist`: logical, whether to attempt to generate and apply a greylist if none is present when running `dba.analyze`.

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

peakFormat — if a sampleSheet is specified, the default peak file format that will be used if the PeakFormat column is absent.

scoreCol — if a sampleSheet is specified, the default column in the peak files that will be used for scoring if the ScoreCol column is absent.

bLowerScoreBetter — if a sampleSheet is specified, the sort order for peak scores if the LowerBetter column is absent.

filter — if a sampleSheet is specified, a filter value if the Filter column is absent. Peaks with scores lower than this value (or higher if bLowerScoreBetter or LowerBetter is TRUE) will be removed.

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 sampleSheet.

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.

bSummarizedExperiment — logical indicating whether to return resulting object as a SummarizedExperiment.

bCorPlot — logical indicating that a correlation heatmap should be plotted before returning. If DBA is NULL (a new DBA object is being created), and bCorPlot is missing, then this will take the default value (FALSE). However if DBA is NULL (a new DBA object is being created), and bCorPlot is specified, then the specified value will become the default value of bCorPlot for the resultant DBA object.

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

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

dir — Directory path. If supplied, files referenced in the sampleSheet will have this path prepended. Applies to PeakFiles, bamReads, bamControl, and Spikein, if present. If sampleSheet is a filepath, this will prepended to that as well.

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)`

MODE: Convert a DBA object to a SummarizedExperiment object:

`dba(DBA, bSummarizedExperiment=TRUE)`

Value

DBA object

Author(s)

Rory Stark and Gordon Brown

See Also

dba.peakset, dba.show, DBA.config.

Examples

```
# Create DBA object from a samplesheet
## Not run:
basedir <- system.file("extra", package="DiffBind")
tamoxifen <- dba(sampleSheet="tamoxifen.csv", dir=basedir)
tamoxifen

tamoxifen <- dba(sampleSheet="tamoxifen_allfields.csv")
tamoxifen

tamoxifen <- dba(sampleSheet="tamoxifen_allfields.csv",config="config.csv")
tamoxifen

## End(Not run)

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

# change peak caller but leave peak format the same
basedir <- system.file("extra", package="DiffBind")
tamoxifen <- dba(sampleSheet="tamoxifen.csv", dir=basedir,
                 peakCaller="macs", peakFormat="raw", scoreCol=5 )
dba.show(tamoxifen, attributes=c(DBA_TISSUE,DBA_CONDITION,DBA_REPLICATE,DBA_CALLER))

# Convert DBA object to SummarizedExperiment
data(tamoxifen_counts)
sset <- dba(tamoxifen,bSummarizedExperiment=TRUE)
```

sset

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 from the DiffBind package.

DBA objects are usually constructed using the dba function.

There are a number of internal parameters that can be set, and defaults overridden, by setting DBA$config options:

- DBA$config$AnalysisMethod: either DBA_DESEQ2 or DBA_EDGER.
- DBA$config$th: default threshold for reporting and plotting analysis results.
- DBA$config$DataType: default class for peaks and reports (DBA_DATA_GRANGES, DBA_DATA_RANGEDDATA, or DBA_DATA_FRAME).
- DBA$config$RunParallel: logical indicating if counting and analysis operations should be run in parallel using multicore by default.
- DBA$config$cores: number of cores to use when performing multi-core parallel processing.
- DBA$config$minQCth: numeric, for filtering reads based on mapping quality score; only reads with a mapping quality score greater than or equal to this will be counted.
- DBA$config$fragmentSize: numeric indicating mean fragment size for single-end counting. Reads will be extended to this length before counting overlaps. May be a vector of lengths, one for each sample.

- DBA$config$bCorPlot: logical indicating that a correlation heatmap should be plotted automatically
- DBA$config$ReportInit: string to append to the beginning of saved report file names.
- DBA$config$bUsePval: logical, default indicating whether to use FDR (FALSE) or p-values (TRUE).
- DBA$config$doBlacklist: logical, whether to attempt to find and apply a blacklist if none is present when running dba.analyze.
- DBA$config$doGreylist logical, whether to attempt to generate and apply a greylist if none is present when running dba.analyze.
- DBA$config$DataType The class of object for returned reports and peaksets:
  - DBA_DATA_GRANGES
  - DBA_DATA_RANGEDDATA
  - DBA_DATA_FRAME
  - DBA_DATA_SUMMARIZED_EXPERIMENT
- DBA$config$mergeOverlap: The overlap (in basepairs) between peaks to merge when generating a consensus peakset. A positive valuecontrols how many basepairs peaks must overlap to be merged, while a negative value will result in non-overlapping peaks to be merged,

  If absent, the default value of 1 will result in any peaks overlapping by at least one basepair to be merged into a single interval.
- DBA$config$design: When calling dba.contrast, if design parameter is missing, this will be used as the value for that parameter.
- DBA$config$edgeR$bTagwise: logical indicating if edgeR::estimateGLMTagwiseDisp should be called when performing an edgeR analysis. If absent the default is TRUE, so setting this to FALSE prevents the tagwise dispersion estimate form being calculated.
- DBA$config$DESeq2$fitType: logical indicating the fitType to be used in DESeq2::estimateDispersions when performing a DESeq2 analysis. If absent the default is local.
- DBA$config$savePrefix: When calling dba.save or dba.load, this value (if present) will override the default value for the pre parameter.
- DBA$config$saveExt: When calling dba.save or dba.load, this value (if present) will override the default value for the ext parameter.
- DBA$config$greylist.pval: pvalue cutoff to use when generating a greylist using GreyListChIP::calcThreshold. If missing, the default is 0.999
- DBA$config$saveExt: When calling dba.save, this value (if present) will override the default value for the ext parameter.
- DBA$config$yieldSize: yieldSize indicating how many reads to process at one time; default is 5000000. The lower this value, the less memory will be used, but the more time it will take to complete the count operation.
- DBA$config$intersectMode: mode indicating which overlap algorithm to use; default is "IntersectionNotEmpty"
- DBA$config$singleEnd: logical indicating if reads are single end; if NULL, status will be automatically detected.
- DBA$config$fragments: logical indicating how unmatched reads are counted; default is FALSE.

- DBA$config$scanbamparam: ScanBamParam object to pass to summarizeOverlaps. If present, bRemoveDuplicates is ignored.
- DBA$config$pp.style: Sets style parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$pp.nOfWindows: Sets nOfWindow parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$bin_size: Sets bin_size parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$distanceAround: Sets distanceAround parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$distanceUp: Sets distanceUp parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$distanceDown: Sets distanceDown parameter for profileplyr::BamBigwig_to_chipProfile when calling dba.plotProfile.
- DBA$config$id: character string to use to replace "ID" when displaying a DBA object (dba.show)
- DBA$config$group: character string to use to replace "Group" when displaying a DBA object (dba.show)
- DBA$config$tissue: character string to use to replace "Tissue" when displaying a DBA object (dba.show)
- DBA$config$factor: character string to use to replace "Factor" when displaying a DBA object (dba.show)
- DBA$config$condition: character string to use to replace "Condition" when displaying a DBA object (dba.show)
- DBA$config$treatment: character string to use to replace "Treatment" when displaying a DBA object (dba.show)
- DBA$config$replicate: character string to use to replace "Replicate" when displaying a DBA object (dba.show)
- DBA$config$caller: character string to use to replace "Caller" when displaying a DBA object (dba.show)
- DBA$config$reads: character string to use to replace "Reads" when displaying a DBA object (dba.show)

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)

data(tamoxifen_greylist)
```

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. Also includes background bins counts for background normalization.
`tamoxifen_analysis`	load tamoxifen resistance dataset DBA object with count (affinity) data and `DESeq2`-based differential binding analysis results. This analysis uses a blacklists, computed greylists, background normalization, and a two-factor design.
`tamoxifen_greylist`	load greylist for tamoxifen dataset. Generated as shown in `dba.blacklist` example: `dba.blacklist`.

Details

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

Data used to create these objects can be downloaded at https://content.cruk.cam.ac.uk/bioinformatics/software/DiffBind/DiffBind_vignette_data.tar.gz.

Value

loads a DBA object named `tamoxifen` (or `tamoxifen.greylist`).

Note

The script for generating these files (`GenerateDataFiles.R`) is included with the package in the `inst/extras` directory.

Author(s)

Rory Stark

Examples

```
data(tamoxifen_peaks)
tamoxifen
data(tamoxifen_counts)
plot(tamoxifen)
data(tamoxifen_analysis)
dba.plotMA(tamoxifen)
data(tamoxifen_greylist)
tamoxifen.greylist$master
```

`dba.analyze` *Perform differential binding affinity analysis*

Description

Performs differential binding affinity analysis. Performs default generation of a consensus peakset, read counting, normalization, and setting up of contrasts if they have not been specified.

Usage

```
dba.analyze(DBA, method=DBA$config$AnalysisMethod, design,
            bBlacklist=DBA$config$doBlacklist,
            bGreylist=DBA$config$doGreylist,
            bRetrieveAnalysis=FALSE, bReduceObjects=TRUE,
            bParallel=DBA$config$RunParallel)
```

Arguments

`DBA`	Either a `DBA` object, or a sample sheet (either a `character` vector with the name of the sample sheet, or a `data.frame` containing the experimental metadata. If no blacklist or greylists are included, a call will be made to `dba.blacklist` using defaults. This can be skipped by setting the `bBlacklist` and/or `bGreylist` parameters. If no counts are included, a default consensus will be formed and read counts computed via a call to `dba.count` using defaults. If no normalization has been specified, the reads will be normalized via a call to `dba.normalize` using defaults. If no contrasts are specified (`DBA$contrast` is `NULL`), default contrasts will be added via a call to `dba.contrast` using defaults.
`method`	Underlying method, or vector of methods, by which to analyze differential binding affinity. Supported methods:

- DBA_EDGER use edgeR package for analysis
- DBA_DESEQ2 use DESeq2 package for analysis
- DBA_ALL_METHODS perform two analyses, using both edgeR and DESeq2

design — If present and a character string, will be used as the design formula for the analysis, replacing any previously established design if present.

If FALSE, will complete analysis in pre-version 3 mode.

See link{dba.contrast}.

bBlacklist — If TRUE, and no blacklist has been applied to the DBA object (or when starting from a samplesheet), the read bam files will be examined to determine the reference genome, and an appropriate blacklist applied, if available. See link{dba.blacklist}.

bGreylist — If TRUE, and no greylist has been applied to the DBA object (or when starting from a samplesheet), the control bam files, if present, will be examined to determine the reference genome, greylists will be computed for each, merged into a master greylist, and applied to the peaksets. See link{dba.blacklist}.

bRetrieveAnalysis — If changed from FALSE, the underlying DE analysis object is returned instead of running a new analysis. Possible values for bRetrieveAnalysis:

- DBA_DESEQ2 Returns the DESeq2 DESeqDataSet.
- DBA_EDGER Returns the edgeR DGEList.
- TRUE Returns the DESeq2 DESeqDataSet, if present. If not, returns the edgeR DGEList, if present..

An analysis object will only be successfully returned if there is at least one contrast utilizing an explicit design (see dba.contrast), and an analysis has been carried out.

bReduceObjects — logical indicating whether strip the analysis objects of unnecessary fields to save memory. If it is desired to use the DBA$contrasts[[n]]$edgeR and/or DBA$contrasts[[n]]$DESeq2 objects directly in the edgeR and/or DESeq2 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

In general, prior to calling dba.analyze, dba.count should have been run. If no contrasts have been established prior to invoking dba.analyze, then the default set of contrasts will be added using (dba.contrast).

If no normalization parameters have been supplied by calling dba.normalize, default normalization parameters will be used.

See the DBA User Guide for more details on how the edgeR and DESeq2 analyses are carried out.

Value

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

Alternatively, an analysis object (either a DESeqDataSet or a DGEList) if bRetrieveAnalysis if not FALSE.

Note

If there is a blocking factor for the contrast(s) specified using a previous call to dba.contrast with design=FALSE, a multi-factor analysis will automatically be carried out in addition to a single factor analysis.

Author(s)

Rory Stark

See Also

dba.blacklist, dba.count, dba.contrast, dba.normalize, dba.report, DBA.config.

Examples

```
data(tamoxifen_counts)
dba.analyze(tamoxifen)

tamoxifen <- dba.analyze(tamoxifen, method=DBA_ALL_METHODS,
                         design="~Tissue + Condition")
dba.show(tamoxifen, bContrasts=TRUE)

dba.analyze(tamoxifen, bRetrieveAnalysis=TRUE)
edger.object <- dba.analyze(tamoxifen, bRetrieveAnalysis=DBA_EDGER)
class(edger.object)
```

---

dba.blacklist *Apply blacklists and/or greylists to peaks (and generate greylist)*

Description

Filters peak intervals that overlap a blacklist (from ENCODE or user supplied.) Filter peak intervals that overlap a greylist, either user supplied or generated based on aligned reads for control samples (e.g. Inputs).

Usage

```
dba.blacklist(DBA, blacklist=DBA$config$doBlacklist,
              greylist=DBA$config$doGreylist,
              Retrieve, cores=DBA$config$cores)
```

Arguments

DBA	DBA object
blacklist	If not equal to FALSE, specifies that a blacklist should be applied to the peak intervals in the DBA object. If equal to TRUE, the read bam files will be examined to determine an appropriate reference genome. If successful, and a blacklist is available for that genome, it will be applied. A user specified blacklist can be specified by setting this parameter to a GRanges object containing the blacklisted regions. Otherwise, this parameter may be set to one of the following constants, indicating which of the ENCODE blacklists should be applied: • DBA_BLACKLIST_HG19: Homo sapiens 19 (chromosomes have "chr") • DBA_BLACKLIST_HG38: Homo sapiens 38 (chromosomes have "chr") • DBA_BLACKLIST_GRCH37: Homo sapiens 37 (chromosomes are numbers) • DBA_BLACKLIST_GRCH38: Homo sapiens 38 (chromosomes are numbers) • DBA_BLACKLIST_MM9: Mus musculus 9 • DBA_BLACKLIST_MM10: Mus musculus 10 • DBA_BLACKLIST_CE10: C. elegans 10 • DBA_BLACKLIST_CE11: C. elegans 11 • DBA_BLACKLIST_DM3: Drosophila melanogaster 3 • DBA_BLACKLIST_DM6: Drosophila melanogaster 6
greylist	If not equal to FALSE, specifies that a greylist should be applied to the peak intervals in the DBA object. If equal to TRUE, the control bam files (if present), will be examined to determine an appropriate reference genome. Genomes associated with a valid BSgenome can be detected. If successful, this genome will be used to generate greylists for each available control (eg specified as bamControls in the sample sheet.)). The greylist parameter can also be set explicitly to either a valid BSgenome object, or a character string with the name of a valid BSgenome object. The following constants map to a subset of possible BSgenome objects: • DBA_BLACKLIST_HG19 : seqinfo from BSgenome.Hsapiens.UCSC.hg19 • DBA_BLACKLIST_HG38 : seqinfo from BSgenome.Hsapiens.UCSC.hg38 • DBA_BLACKLIST_GRCH38: seqinfo from BSgenome.Hsapiens.NCBI.GRCh38 • DBA_BLACKLIST_MM9 : seqinfo from BSgenome.Mmusculus.UCSC.mm9 • DBA_BLACKLIST_MM10 : seqinfo from BSgenome.Mmusculus.UCSC.mm10 • DBA_BLACKLIST_CE10 : seqinfo from BSgenome.Celegans.UCSC.ce10 • DBA_BLACKLIST_CE11 : seqinfo from BSgenome.Celegans.UCSC.ce11 • DBA_BLACKLIST_DM3 : seqinfo from BSgenome.Dmelanogaster.UCSC.dm3 • DBA_BLACKLIST_DM6 : seqinfo from BSgenome.Dmelanogaster.UCSC.dm6 A user specified greylist can also be specified by setting this parameter to a GRanges object containing the greylisted regions. It can also be a list with an element named greylist$master, which is a GRanges object containing the greylist to be applied.

Retrieve — If present, some aspects of a previous run of the function is retrieved instead of returning a DBA object.

If Retrieve=DBA_BLACKLIST, the blacklist, if present, is returned as a GRanges object.

If Retrieve=DBA_GREYLIST, the greylist, if present, is returned. If it was generated from more than one control, it will be returned as a list object with the first element (named $master) a GRanges object containing the merged greylist, and the second element (named $controls) being a GRangesList with each element containing the greylist for one control

If Retrieve=DBA_BLACKLISTED_PEAKS, the excluded peaks for each sample will be returned in a GRangesList object (with each element containing the filtered peak intervals for each sample). If counts are available for the peaks, this will include the following metadata columns:

- cReads: Number of control reads overlapping this interval
- Reads: Number of primary (ChIP) reads overlapping this interval
- Score: Read score calculated by dba.count

Note that the if Retrieve is set, dba.blacklist must have been previously run, and all other parameters will be ignored.

cores — Parallel cores to use when running greylist generation.

Details

This function is intended to filter peak intervals that fall in regions of the genome that are known to be problematic for ChIP analysis. Blacklists, which are derived for a reference genome and should be applied for any experiments that use that reference, are distinguished from greylists, which are derived on a per-experiment basis using anomalous pileups in the control tracks (such as Inputs).

A core set of blacklists have been defined as part of the ENCODE project (see references).

Greylists can be generated using this function, which serves as a front-end to the GreyListChIP package. See the details of that package for more information on how it works. Note that the GreyListChIP package can be utilized separately to generate greylists with more fine-grained control, with the results passed back to DiffBind to filter peaks.

Value

DBA object, with peaks filtered (unless Retrieve is specified.)

Note

The p threshold can be altered by setting DBA$config$greylist.pval. The default is 0.999. See GreyListChIP::calcThreshold for details.

Ideally, Blacklists and Greylists will be applied to the aligned reads prior to calling peaks, as removing reads in anomalous regions will yield better background noise models. Once greylists have been generated, peaks can be re-called and read into DiffBind.

Author(s)

Rory Stark with thanks to Gord Brown

References

- Amemiya HM, Kundaje A, Boyle AP. The ENCODE blacklist: identification of problematic regions of the genome. Sci Rep. 2019 Dec; 9(1) 9354 DOI: 10.1038/s41598-019-45839-z
- ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012 Sep 6;489(7414):57-74. doi: 10.1038/nature11247.
- Brown, Gord. Generating Grey Lists from Input Libraries. Bioconductor. https://bioconductor.org/packages/release/bio

See Also

GreyListChIP (GreyList), BSgenome, DBA.config.

Examples

```
data(tamoxifen_peaks)
## Not run: tamoxifen <- dba.blacklist(tamoxifen, blacklist=TRUE,
                                       greylist="BSgenome.Hsapiens.UCSC.hg19")
## End(Not run)
data(tamoxifen_greylist)
tamoxifen <- dba.blacklist(tamoxifen, blacklist=DBA_BLACKLIST_HG19,
                           greylist=tamoxifen.greylist$master)
dba.blacklist(tamoxifen,Retrieve=DBA_GREYLIST)

data(tamoxifen_counts)
tamoxifen <- dba.count(tamoxifen, peaks=NULL, score=DBA_SCORE_CONTROL_READS)
tamoxifen <- dba.blacklist(tamoxifen, blacklist=DBA_BLACKLIST_HG19,
                           greylist=tamoxifen.greylist$master)
blacklisted <- dba.blacklist(tamoxifen, Retrieve=DBA_BLACKLISTED_PEAKS)
mean(blacklisted[[1]]$cReads)
mean(dba.peakset(tamoxifen,peaks=1,bRetrieve=TRUE)$Score)
```

---

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

Description

Sets up contrasts for differential binding affinity analysis

Usage

```
dba.contrast(DBA, design=missing(group1), contrast,
             group1, group2=!group1, name1, name2,
             minMembers=3, block, bNot=FALSE, bComplex=FALSE,
             categories=c(DBA_TISSUE,DBA_FACTOR,DBA_CONDITION,DBA_TREATMENT),
             bGetCoefficients=FALSE, reorderMeta)
```

Arguments

DBA	DBA object with count data
design	Either a logical value, or a character string containing a valid design formula. If a logical value is specified, TRUE indicates that a design should automatically be generated. If contrast is missing, contrasts will automatically be added and an appropriate design computed. If a contrast is specified, it must consist of a character vector of length three, containing a factor and two factor values. No groups can be specified. If set to FALSE, the contrast will be added between the groups, if specified; otherwise, if group is missing, all possible contrasts will be added. If a design formula is specified, it must be composed from the following allowable factors: • Tissue • Factor • Condition • Treatment • Replicate • Caller

If design is not explictly specified, and no group is specified, then design will be set to the value of DBA$config$design, if present (see DiffBind3).

contrast	If a design has been specified (previously or in the current call), the following contrasts forms may be indicated: • Character vector of length three. The first element is a factor from the design. The second and third elements are values for that factor associated with sample groups. • List of length 1, containing a design matrix column name (as obtained using bGetCoefficients). • List of length 2, containing two design matrix column names (as obtained using bGetCoefficients), first the numerator and the second the denominator. • Character vector of length one, containing a design matrix column name (as obtained using bGetCoefficients). • Numeric vector of the same length as the list of design matrix column names (as obtained using bGetCoefficients), with a weighting for each column.
group1	mask of samples in first group (when adding a specific contrast). See dba.mask. Can not be used with an explicit design.
group2	mask of samples in second group (when adding a specific contrast). See dba.mask. Can not be used with an explicit design.
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.

`bNot` include contrasts consisting of a group and all other samples not in that group (indicated by a ! in the contrast name).

`bComplex` include complex contrasts where groups include samples with the same values for multiple factors.

`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 factor value (true), while any peaksets not included in the vector take the alternative factor 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 peakset should appear at most once, and any peaksets not specified will be given an default value (other).

`bGetCoefficients`

If `TRUE`, return the names of the columns (coefficients) associated with the `design`. These can be used to specify a `contrast`. If `bGetCoefficients=TRUE`, all other parameters (except `DBA` and `design`, if specified) will be ignored.

`reorderMeta` By default, the metadata factor levels will be ordered in the order they appear in the sample sheet. They can be re-ordered using this parameter. `reorderMeta` is specified as a list, with each element being a vector of character strings corresponding to unique factor values in the desired order. Each element should be named for the appropriate metadata factor, one of:

- Tissue
- Factor
- Condition
- Treatment
- Replicate
- Caller

If the vector of factor values contains a subset of the possible values, the specified values will be set to be ordered first, with the remaining values following in their default order. If only one factor value is supplied, it will be set as the reference (or "control") value. Contrasts that are no longer valid will be removed (and a warning issued) if detected. These include contrasts specified as a numeric vector of coefficients, or contrasts specified using coefficient names that no longer exists after reordering the metadata factor levels. Any existing analysis will be removed when metadata factor levels are reordered, necessitating another call to `dba.analyze`

Details

MODE: Set up a specific contrast using a design:

```
dba.contrast(DBA, design, contrast)
```

MODE: Set up all possible contrasts:

```
dba.contrast(DBA, minMembers, categories)
```

MODE: Set up a specific contrast without an explicit design:

```
dba.contrast(DBA, design=FALSE, 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=TRUE)`.

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

See Also

`dba.analyze`, `DBA.config`.

Examples

```
# Set up an explicit contrast
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen, contrast=c("Condition","Responsive","Resistant"))
tamoxifen
tamoxifen <- dba.analyze(tamoxifen)
dba.show(tamoxifen,bContrasts=TRUE)

# Add another contrast
tamoxifen <- dba.contrast(tamoxifen, contrast=c("Tissue","MCF7","BT474"))
dba.show(tamoxifen,bDesign=TRUE)

# Change design
tamoxifen <- dba.contrast(tamoxifen,design="~Tissue + Condition")
tamoxifen <- dba.analyze(tamoxifen)
tamoxifen

# Automatically add all contrasts between sample groups
# where at least THREE samples have the same factor value
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen)
tamoxifen

# Automatically add all contrasts between sample groups
# where at least TWO samples have the same factor value
tamoxifen <- dba.contrast(tamoxifen, minMembers=2)
dba.show(tamoxifen,bContrasts=TRUE)

### Use of complex contrasts
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen, contrast=c("Tissue","BT474","MCF7"))
dba.contrast(tamoxifen, bGetCoefficients=TRUE)

#Change design and factor ordering
tamoxifen <- dba.contrast(tamoxifen,design="~Tissue + Condition",
                          reorderMeta=list(Condition="Responsive",
                          Tissue=c("MCF7","ZR75","T47D","BT474")))
dba.contrast(tamoxifen, bGetCoefficients=TRUE)
tamoxifen <- dba.contrast(tamoxifen,contrast="Tissue_BT474_vs_MCF7")
tamoxifen <- dba.contrast(tamoxifen,contrast=list("Tissue_BT474_vs_MCF7"))
tamoxifen <- dba.contrast(tamoxifen,contrast=c(0,0,0,1,0))
tamoxifen <- dba.contrast(tamoxifen,
                          contrast=list("Tissue_BT474_vs_MCF7","Tissue_T47D_vs_MCF7"))
tamoxifen <- dba.contrast(tamoxifen,contrast=c(0,0,-1,1,0))
tamoxifen <- dba.contrast(tamoxifen,contrast=c(0,0,0,0,1))
dba.show(tamoxifen,bContrasts=TRUE)
tamoxifen <- dba.analyze(tamoxifen)
tamoxifen
tamoxifen <- dba.contrast(tamoxifen,
                          contrast=c("Condition","Responsive","Resistant"))
tamoxifen <- dba.analyze(tamoxifen)
dba.show(tamoxifen,bContrasts=TRUE)[7:8,]
```

```
dba.plotVenn(tamoxifen, contrast=7:8, bDB=TRUE,
             bAll=FALSE, bGain=TRUE, bLoss=TRUE)

## Explicit contrast, without design
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen, design=FALSE,
                          group1=tamoxifen$masks$Responsive, name1="Responsive",
                          group2=tamoxifen$masks$Resistant,  name2="Resistant",
                          block=DBA_TISSUE)
dba.show(tamoxifen, bContrasts=TRUE)
tamoxifen <- dba.analyze(tamoxifen)
dba.show(tamoxifen,bContrasts=TRUE)
dba.plotVenn(tamoxifen,contrast=1,method=c(DBA_DESEQ2,DBA_DESEQ2_BLOCK))
```

`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_NORMALIZED,
          fragmentSize=DBA$config$fragmentSize,
          summits=200, filter=1, bRemoveDuplicates=FALSE, bScaleControl=TRUE,
          bSubControl = is.null(DBA$greylist),
          mapQCth=DBA$config$mapQCth, filterFun=max, minCount=0,
          bLog=FALSE, bUseSummarizeOverlaps=TRUE,
          readFormat=DBA_READS_DEFAULT, bParallel=DBA$config$RunParallel)
```

Arguments

`DBA`	DBA object
`peaks`	If GRanges, RangedData, dataframe, or matrix, this parameter contains the intervals to use for counting. If character string, it specifies a file containing the intervals to use (with the first three columns specifying chromosome, startpos, endpos).If missing or a mask, generates a consensus peakset using minOverlap parameter (after applying the mask if present). If NULL, the `score`, `filter`, and `summits` parameters are honored, updating the global binding matrix without re-counting in the cases of `score` and `filter`, and only counting after re-centering in the case of `summits`.
`minOverlap`	only include peaks in at least this many peaksets when generating consensus peakset (i.e. when peaks parameter is missing). If minOverlap is between zero and one, peak will be included from at least this proportion of peaksets.

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_NORMALIZED	normalized reads, as set by dba.normalize
DBA_SCORE_READS	raw read count for interval using only reads from ChIP
DBA_SCORE_CONTROL_READS	raw read count for interval using only reads from Control
DBA_SCORE_READS_FOLD	raw read count for interval from ChIP divided by read count for interval from control
DBA_SCORE_READS_MINUS	raw read count for interval from ChIP minus read count for interval from control
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_RPKM_MINUS	RPKM for interval from ChIP minus RPKM for interval from control
DBA_SCORE_SUMMIT	summit height (maximum read pileup value)
DBA_SCORE_SUMMIT_ADJ	summit height (maximum read pileup value), normalized to relative library size
DBA_SCORE_SUMMIT_POS	summit position (location of maximum read pileup)

If DBA is a report-based object, the allowable scores are:

DBA_SCORE_FOLD	log2 Fold Change
DBA_SCORE_CONCENTRATION	mean concentration (log2)
DBA_SCORE_CONC_NUMERATOR	mean concentration (log2) of first group in contrast
DBA_SCORE_CONC_DENOMINATOR	mean concentration (log2) of second group in contrast
DBA_SCORE_PVAL	p-value
DBA_SCORE_FDR	FDR

fragmentSize — 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 set to zero, the read size indicated in the BAM/BED file will be used. fragmentSize may also be a vector of values, one for each ChIP sample plus one for each unique Control library.

summits — unless set to FALSE, summit heights (read pileup) and locations will be calculated for each peak. The values can retrieved using dba.peakset. The summits can also be used as a read score in the global binding matrix (see score).

If the value of summits is TRUE (or 0), the summits will be calculated but the peaksets will be unaffected. If the value is greater than zero, all consensus peaks will be re-centered around a consensus summit, with the value of summits indicating how many base pairs to include upstream and downstream of the summit (so all consensus peaks will be of the same width, namely 2 * summits + 1).

Note that if summits is greater than zero, the counting procedure will take twice as long.

filter — value to use for filtering intervals with low read counts. The filterFun will be applied to the counts for each interval, and if it returns a value below the filter value, the interval will be removed from further analysis. If peaks is NULL, will remove sites from existing DBA object without recounting. If filter is a vector of values, dba.count will return a vector of the same length, indicating how many intervals will be retained for each specified filter level.

NB: the filtering will be based on RPKM values. If bSubControl is FALSE, this is the RPKM value of the read counts (equivalent to score=DBA_SCORE_RPKM. If bSubControl is TRUE, this is the RPKM value of the control counts subtracted from the RPKM of the read counts (equivalent to score=DBA_SCORE_RPKM_MINUS).

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. Note that if bLowMem is set, duplicates needs to have been already marked in all of the BAM files. The built-in counting code may not correctly handle certain cases when the bRemoveDuplicates parameter is set to TRUE. These cases include paired-end data and datasets where the read length may differ within a single BAM file. In these cases, see the bUseSummarizeOverlaps parameter.

bScaleControl
: logical indicating if the Control reads should be scaled based on relative library sizes. If TRUE, and there are more reads in the Control library than in the ChIP library, the number of Control reads for each peak will be multiplied by a scaling factor determined by dividing the total number of reads in the ChIP library by the total number of reads in the Control library. If this value is not an integer, the number of Control reads for each peak will be the next highest integer.

bSubControl
: logical indicating whether Control read counts are subtracted for each site in each sample. If there are more overlapping control reads than ChIP reads, the count will be set to the minCount value specified when dba.count was called, or zero if no value is specified.

  If bSubControl is not explicitly specified, it will be set to TRUE unless a greylist has been applied (see dba.blacklist).

mapQCth
: for filtering by mapping quality (mapqc). Only alignments with mapping scores of at least this value will be included. Only applicable for bam files when bUseSummarizeOverlaps=FALSE (setting DBA$config$scanbamparam appropriately to filter on quality scores when using summarizeOverlaps.)

filterFun
: function that will be invoked for each interval with a vector of scores for each sample. Returns a score that will be evaluated against the filter value (only intervals with a score at least as high as filter will be kept). Default is max, indicating that at least one sample should have a score of at least filter; other useful values include sum (indicating that all the scores added together should be at least filter) and mean (setting a minimum mean normalized count level). Users can supply their own function as well.

minCount
: minimum read count value. Any interval with fewer than this many overlapping reads will be set to have this count. Also applies to scores.

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

bUseSummarizeOverlaps
: logical indicating that summarizeOverlaps should be used for counting instead of the built-in counting code. This option is slower but uses the more standard counting function. If TRUE, all read files must be BAM (.bam extension), with associated index files (.bam.bai extension). The fragmentSize parameter must absent.

See notes for when the bRemoveDuplicates parameter is set to TRUE, where the built-in counting code may not correctly handle certain cases and bUseSummarizeOverlaps should be set to TRUE.

Five additional parameters for summarizeOverlaps may be specified in DBA$config:

DBA$config$yieldSize	yieldSize indicating how many reads to process at one time; default is 5000000. The lower th
DBA$config$intersectMode	mode indicating which overlap algorithm to use; default is "IntersectionNotEmpty"
DBA$config$singleEnd	logical indicating if reads are single end; if NULL, status will be automatically detected.
DBA$config$fragments	logical indicating how unmatched reads are counted; default is FALSE
DBA$config$inter.feature	logical indicating the setting for the inter.feature parameter; default is TRUE
DBA$config$scanbamparam	ScanBamParam object to pass to summarizeOverlaps. If present, bRemoveDuplicates is ign

readFormat — Specify the file type of the read files, over-riding the file extension. Possible values:

DBA_READS_DEFAULT	use file extension (.bam, .bed, .bed.gz) to determine file type
DBA_READS_BAM	assume the file type is BAM, regardless of the file extension
DBA_READS_BED	assume the file type is BED (or zipped BED), regardless of the file extension.

Note that if readFormat is anything other than DBA_READS_DEFAULT, all the read files must be of the same file type.

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

See Also

dba.analyze

Examples

```
# These won't run unless you have the reads available in a BAM 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

#First make consensus peaksets from each set of replicates,
```

```
#then derive master consensus set for counting from those
data(tamoxifen_peaks)
tamoxifen <- dba.peakset(tamoxifen,consensus = -DBA_REPLICATE)
## Not run: tamoxifen <- dba.count(tamoxifen, peaks=tamoxifen$masks$Consensus)
tamoxifen

# Change binding affinity scores
data(tamoxifen_counts)
dba.peakset(tamoxifen, bRetrieve=TRUE) # default: DBA_SCORE_NORMALIZED
tamoxifen <- dba.count(tamoxifen,peaks=NULL,score=DBA_SCORE_READS)
dba.peakset(tamoxifen, bRetrieve=TRUE)
tamoxifen <- dba.count(tamoxifen,peaks=NULL,score=DBA_SCORE_RPKM_MINUS)
dba.peakset(tamoxifen, bRetrieve=TRUE)

# Plot effect of a range of filter values and then apply filter
data(tamoxifen_counts)
rate.max <- dba.count(tamoxifen, peaks=NULL, filter=0:250)
rate.sum <- dba.count(tamoxifen, peaks=NULL, filter=0:250,filterFun=sum)
plot(0:250,rate.max/rate.max[1],type='l',xlab="Filter Value",ylab="Proportion Retained Sites")
lines(0:250,rate.sum/rate.sum[1],col=2)
tamoxifen <- dba.count(tamoxifen,peaks=NULL,filter=125,filterFun=sum)
tamoxifen

# Calculate summits
data(tamoxifen_counts)
# pre-counted with summits=250 or 501bp intervals
as.numeric(dba.show(tamoxifen)$FRiP)
## Not run: tamoxifen <- dba.count(tamoxifen,peaks=NULL,summits=50)
# re-counted with summits=50 or 101bp intervals
as.numeric(dba.show(tamoxifen)$FRiP)
```

`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

See Also

dba.save, DBA.config.

Examples

```
data(tamoxifen_peaks)
savefile <- dba.save(tamoxifen,'tamoxifenPeaks')
savefile
rm(tamoxifen)
tamoxifen <- dba.load('tamoxifenPeaks')
tamoxifen
unlink(savefile)
```

---

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 specifies 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

See Also

dba.show

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.normalize *Specify parameters for normalizing a dataset; calculate library sizes and normalization factors.*

Description

Enables normalization of datasets using a variety of methods, including background, spike-in, and parallel factor normalization. Alternatively, allows a user to specify library sizes and normalization factors directly, or retrieve computed ones.

Usage

```
dba.normalize(DBA, method = DBA$config$AnalysisMethod,
              normalize = DBA_NORM_DEFAULT, library = DBA_LIBSIZE_DEFAULT,
              background = FALSE, spikein = FALSE, offsets = FALSE,
              libFun=mean, bRetrieve=FALSE, ...)
```

Arguments

DBA	DBA object that includes count data for a consensus peakset.
method	Underlying method, or vector of methods, for which to normalize. Supported methods: • DBA_EDGER use edgeR package for analysis • DBA_DESEQ2 use DESeq2 package for analysis • DBA_ALL_METHODS normalize for both both edgeR and DESeq2

normalize — Either user-supplied normalization factors in a numeric vector, or a specification of a method to use to calculate normalization factors.

Methods can be specified using one of the following:

- DBA_NORM_RLE ("RLE") RLE normalization (native to DBA_DESEQ2, and available for DBA_EDGER).
- DBA_NORM_TMM ("TMM") TMM normalization (native to DBA_EDGER, and available for DBA_DESEQ2).
- DBA_NORM_NATIVE ("native") Use native method based on method: DBA_NORM_RLE for DBA_DESEQ2 or DBA_NORM_TMM for DBA_EDGER.
- DBA_NORM_LIB ("lib") Normalize by library size only. Library sizes can be specified using the library parameter. Normalization factors will be calculated to give each equal weight in a manner appropriate for the analysis method. See also the libFun parameter, which can be used to scale the normalization factors for DESeq2.
- DBA_NORM_DEFAULT ("default") Default method: The "preferred" normalization approach depending on method and whether an explicit design is present. See Details below.
- DBA_NORM_OFFSETS ("offsets") Indicates that offsets have been specified using the offsets parameter, and they should be used without alteration.
- DBA_NORM_OFFSETS_ADJUST ("adjust offsets") Indicates that offsets have been specified using the offsets parameter, and they should be adjusted for library size and mean centering before being used in a DBA_DESEQ2 analysis.

library — Either user-supplied library sizes in a numeric vector, or a specification of a method to use to calculate library sizes.

Library sizes can be based on one of the following:

- DBA_LIBSIZE_FULL ("full") Use the full library size (total number of reads in BAM/SAM/BED file)
- DBA_LIBSIZE_PEAKREADS ("RiP") Use the number of reads that overlap consensus peaks.
- DBA_LIBSIZE_BACKGROUND ("background") Use the total number of reads aligned to the chromosomes for which there is at least one peak. This required a background bin calculation (see parameter background). These values are usually the same or similar to DBA_LIBSIZE_FULL.
- DBA_LIBSIZE_DEFAULT ("default") Default method: The "preferred" library size depending on method, background, and whether an explicit design is present. See Details below.

background — This parameter controls the option to use "background" bins, which should not have differential enrichment between samples, as the basis for normalizing (instead of using reads counts overlapping consensus peaks). When enabled, the chromosomes for which there are peaks in the consensus peakset are tiled into large bins and reads overlapping these bins are counted.

If present, background can either be a logical value, a numeric value, or a previously computed $background object.

If background is a logical value and set to TRUE, background bins will be computed using the default bin size of 15000bp. Setting this value to FALSE will prevent background mode from being used in any default settings.

If background is a numeric value, it will be used as the bin size.

If background is a previously computed $background object, these counts will be used as the background. A $background object can be obtained by calling dba.normalize with bRetrieve=TRUE and method=DBA_ALL_METHODS.

After counting (or setting) background bins, both the normalize and library parameters will be used to determine how the final normalization factors are calculated.

If background is missing, it will be set to TRUE if library=DBA_LIBSIZE_BACKGROUND, or if library=DBA_LIBSIZE_DEFAULT and certain conditions are met (see Details below).

If background is not FALSE, then the library size will be set to library=DBA_LIBSIZE_BACKGROUND

spikein

Either a logical value, a character vector of chromosome names, a GRanges object containing peaks for a parallel factor, or a $background object containing previously computed spike-in read counts.

If spikein is a logical value set to FALSE, no spike-in normalization is performed.

If spikein is a logical value set to TRUE, background normalization is performed using spike-in tracks. There must be a spike-in track for each sample. see dba and/or dba.peakset for details on how to include a spike-in track with a sample (eg. by including a Spikein column in the sample sheet.) All chromosomes in the spike-in bam files will be used.

If spikein is a character vector of one or more chromosome names, only reads on the named chromosome(s) will be used for background normalization. If spike-in tracks are available, reads on chromosomes with these names in the spike-in track will be counted. If no spike-in tracks are available, reads on chromosomes with these names in the main bamReads bam files will be counted.

If spikein is a GRanges object containing peaks for a parallel factor, then background normalization is performed counting reads in the spike-in tracks overlapping peaks in this object.

If spikein is a previously computed $background object, these counts will be used as the spikein background. A $background object can be obtained by calling dba.normalize with bRetrieve=TRUE and method=DBA_ALL_METHODS.

Note that if spikein is not FALSE, then the library size will be set to library=DBA_LIBSIZE_BACKGROUND

offsets

This parameter controls the use of offsets (matrix of normalization factors) instead of a single normalization factor for each sample. It can either be a logical value, a matrix, or a SummarizedExperiment.

If it is a logical value and set to FALSE, no offsets will be computed or used. A value of TRUE indicates that an offset matrix should be computed using a loess fit.

Alternatively, user-calculated normalization offsets can be supplied as a matrix or as a SummarizedExperiment (containing an assay named "offsets"). In this case, the user may also set the normalize parameter to indicate whether the offsets should be applied as-is to a DESeq2 analysis (DBA_NORM_OFFSETS, default), or if they should be adjusted for library size and mean centering (DBA_NORM_OFFSETS_ADJUST).

libFun — When normalize=DBA_NORM_LIB, normalization factors are calculated by dividing the library sizes for each sample by a common denominator, obtained by applying libFun to the vector of library sizes.

For method=DBA_EDGER, the normalization factors are further adjusted so as to make all the effective library sizes (library sizes multiplied by normalization factors) the same, and adjusted to multiply to 1.

bRetrieve — If set to TRUE, information about the current normalization will be returned. The only other relevant parameter in this case is the method.

If method=DBA_DESEQ2 or method=DBA_EDGER, a record will be returned including normalization values for the appropriate analysis method. This record is a list consists of the following elements:

- $norm.method A character string corresponding to the normalization method, generally one of the values that can be supplied as a value to normalize.
- $norm.factors A vector containing the computed normalization factors.
- $lib.method A character string corresponding to the value of the method used to calculate the library size, generally one of the values that can be supplied as a value to library.
- $lib.sizes A vector containing the computed library sizes.
- $background If the normalization if based on binned background reads, this field will be TRUE.
- $control.subtract If control reads were subtracted from the read counts, this field will be TRUE.

If method=DBA_ALL_METHODS, the record be a list with one of the above records for each method for which normalization factors have been computed ($DESeq2 and edgeR).

If background bins have been calculated, this will include an element called $background. This element can be passed in as the value to background or spikein to re-use a previously computed set of reads. It contains three subfields:

- $background$binned a SummarizedExperiment object containing the binned counts.
- $background$bin.size a numeric value with the bin size used.
- $background$back.calc character string indicating how the background was calculated (bins, spike-ins, or parallel factor).

If offsets are available, this will include an element called $offsets with two subfields:

- $offsets$offsets a matrix or a SummarizedExperiment object containing the offsets.
- offsets$offset.method a character string indicating the source of the offsets, either "loess" or "user".

... — Extra parameters to be passed to limma::loessFit when computing offsets.

Details

The default normalization parameters are as follows:

- normalize=DBA_NORM_LIB

- library=DBA_LIBSIZE_FULL
- background=FALSE

If background=TRUE, then the default becomes library=DBA_LIBSIZE_BACKGROUND.

If dba.contrast has been used to set up contrasts with design=FALSE (pre-3.0 mode), then the defaults are:

- normalize=DBA_NORM_DEFAULT
- library=DBA_LIBSIZE_FULL
- background=FALSE

In this case, normalize=DBA_NORM_LIB will be set for method=DBA_DESEQ2 for backwards compatibility.

Value

Either a DBA object with normalization terms added, or (if bRetrieve=TRUE), a record or normalization details.

Note

The csaw package is used to compute background bins and offsets based on limma::loessFit.

See the DiffBind vignette for technical details of how this is done, and the csaw vignette for details on background bins and loess offsets can be used to address different biases in ChIP-seq data.

Author(s)

Rory Stark

See Also

dba.count, dba.analyze, dba.save

Examples

```
# load DBA object with counts
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen,design="~Tissue + Condition")

# default normalization: Full library sizes
tamoxifen <- dba.normalize(tamoxifen)
dba.normalize(tamoxifen, bRetrieve=TRUE)
dba.analyze(tamoxifen)

# RLE/TMM using Reads in Peaks
tamoxifen <- dba.normalize(tamoxifen, method=DBA_ALL_METHODS,
                           normalize=DBA_NORM_NATIVE,
                           library=DBA_LIBSIZE_PEAKREADS)
dba.normalize(tamoxifen, method=DBA_DESEQ2, bRetrieve=TRUE)
dba.normalize(tamoxifen, method=DBA_EDGER, bRetrieve=TRUE)
```

```
tamoxifen <- dba.analyze(tamoxifen, method=DBA_ALL_METHODS)
dba.show(tamoxifen,bContrasts=TRUE)
dba.plotVenn(tamoxifen,contrast=1,method=DBA_ALL_METHODS,bDB=TRUE)

# TMM in Background using precomputed background
norm <- dba.normalize(tamoxifen,method=DBA_ALL_METHODS,bRetrieve=TRUE)
tamoxifen <- dba.normalize(tamoxifen, background=norm$background,
                           normalize="TMM", method=DBA_ALL_METHODS)
tamoxifen <- dba.analyze(tamoxifen)
dba.show(tamoxifen,bContrasts=TRUE)
dba.plotMA(tamoxifen)

# LOESS offsets
tamoxifen <- dba.normalize(tamoxifen, method=DBA_ALL_METHODS, offsets=TRUE)
tamoxifen <- dba.analyze(tamoxifen, method=DBA_ALL_METHODS)
dba.show(tamoxifen,bContrasts=TRUE)

par(mfrow=c(3,1))
dba.plotMA(tamoxifen,th=0,bNormalized=FALSE)
dba.plotMA(tamoxifen,method=DBA_DESEQ2)
dba.plotMA(tamoxifen,method=DBA_EDGER)
```

`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,
            contrast, method=DBA$config$AnalysisMethod, th=DBA$config$th,
            bUsePval=DBA$config$bUsePval,
            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, three, or four peaksets may 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

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_DESEQ2
- DBA_DESEQ2_BLOCK
- DBA_EDGER
- DBA_EDGER_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

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, seven peaksets for a A-B-C overlap, and fifteen peaksets for a A-B-C-D overlap:

`inAll`	peaks in all peaksets
`onlyA`	peaks unique to peakset A
`onlyB`	peaks unique to peakset B
`onlyC`	peaks unique to peakset C
`onlyD`	peaks unique to peakset D
`notA`	peaks in all peaksets except peakset A
`notB`	peaks in all peaksets except peakset B
`notC`	peaks in all peaksets except peakset C
`notD`	peaks in all peaksets except peakset D
`AandB`	peaks in peaksets A and B but not in peaksets C or D
`AandC`	peaks in peaksets A and C but not in peaksets B or D
`AandD`	peaks in peaksets A and D but not in peaksets B or C
`BandC`	peaks in peaksets B and C but not in peaksets A or D
`BandD`	peaks in peaksets B and D but not in peaksets A or C
`CandD`	peaks in peaksets C and D but not in peaksets A or B

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

See Also

dba.plotVenn, dba.plotHeatmap

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, peak.format, reads=0, consensus=FALSE,
            bamReads, bamControl, spikein,
            scoreCol, bLowerScoreBetter, filter, counts,
            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 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 to include in the consensus. 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 and empty peakset (zero peaks), set peaks=NA. When adding a set of consensus peaksets: a sample mask or vector of peakset numbers. Sample sets will be derived only from subsets of these peaksets. 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 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). Must be unique for each sample.
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, and peak.format is missing, a default fie format for the caller will be used (see peak.format). Supported values:

- "raw": default peak.format: raw text file
- "bed": default peak.format: bed file
- "narrow": default peak.format: narrowPeaks file
- "macs": default peak.format: MACS .xls file
- "bayes": default peak.format: bayesPeak file
- "tpic": default peak.format: TPIC file
- "sicer": default peak.format: SICER file
- "fp4": default peak.format: FindPeaks v4 file
- "swembl": default peak.format: SWEMBL file
- "csv": default peak.format: comma separated value file
- "report": default peak.format: csv file saved via dba.report

When adding a consensus peakset, a default value (a concatenation of peak caller names) is assigned if this is missing.

peak.format — peak format string. If specified, overrides the default file format for the specified peak caller. Supported formats (with default score column):

- "raw": raw text file file; scoreCol=4
- "bed": bed file; scoreCol=5
- "narrow": narrowPeaks file; scoreCol=8
- "macs": MACS .xls file; scoreCol=7
- "bayes": bayesPeak file; scoreCol=4, filter=0.5
- "tpic": TPIC file; scoreCol=0 (all scores=1)
- "sicer": SICER file; scoreCol=7
- "fp4": FindPeaks v4 file; scoreCol=5
- "swembl": SWEMBL file; scoreCol=4
- "csv": csv file; scoreCol=4
- "report": report file; scoreCol=9, bLowerScoreBetter=T

reads — total number of ChIPed library reads for the peakset being added.

consensus — either the logical value of the consensus attribute when adding a specific peakset (set to TRUE for consensus peaksets generated by dba.peakset), or a metadata attribute or vector of attributes when generating a set of consensus peaksets. In the latter case, a consensus peakset will be added for each set of samples that have the same values for the specified attributes. Alternatively, attributes may be specified proceeded by a negative sign, in which case a consensus peakset will be added for each set of samples that differ only in their values for those attributes. See examples. Allowable attributes:

- DBA_TISSUE; -DBA_TISSUE
- DBA_FACTOR; -DBA_FACTOR
- DBA_CONDITION; -DBA_CONDITION
- DBA_TREATMENT; -DBA_TREATMENT
- DBA_REPLICATE; -DBA_REPLICATE

- DBA_CALLER; -DBA_CALLER

bamReads — file path of the BAM/BED file containing the aligned reads for the peakset being added.

bamControl — file path of the BAM/BED file containing the aligned reads for the control used for the peakset being added.

spikein — file path of the BAM/BED file containing the aligned reads for the spike-ins used for the peakset being added.

scoreCol — peak column to normalize to 0...1 scale when adding a peakset; 0 indicates no normalization

bLowerScoreBetter — Logical indicating that lower scores indicate higher confidence peaks; default is that higher scores indicate better peaks.

filter — Numeric indicating a filter value for peaks. If present, any peaks with a score less than this value (or higher if bLowerScoreBetter==TRUE) will be removed from the peakset.

counts — Used for adding externally computed peak counts. Can be a filename or a dataframe. Can consist of a single column (or vector) with the counts, or two columns, with an ID for each interval in the first column and the counts in the second column, or four columns (chr, start, end, counts). When counts is specified, peaks and related parameters are ignored, and all peaksets in the DBA object must be specified in this way, all with exactly the same number of intervals.

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. If minOverlap is between zero and one, peak will be included from at least this proportion of 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 of 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_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: Add a sets of consensus peaksets bases on sample sets that share or differ in specified attributes

`dba.peakset(DBA, peaks, consensus, 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 GRanges object when retrieving and/or writing a peakset.

Author(s)

Rory Stark

See Also

to add peaksets using a sample sheet, see dba.

`$config$` options are described in DBA.config.

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"),
                   peak.caller="bed", 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"),
                   peak.caller="bed", 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"),
                   peak.caller="bed", 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

#add consensus peaksets for all sample types by combining replicates
data(tamoxifen_peaks)
tamoxifen <- dba.peakset(tamoxifen,consensus = -DBA_REPLICATE)
dba.show(tamoxifen,mask=tamoxifen$masks$Consensus)

#add consensus peaksets for all sample types by (same tissue and condition)
data(tamoxifen_peaks)
tamoxifen <- dba.peakset(tamoxifen,consensus = c(DBA_TISSUE,DBA_CONDITION))
dba.show(tamoxifen,mask=tamoxifen$masks$Consensus)
dba.plotVenn(tamoxifen,tamoxifen$masks$Responsive & tamoxifen$masks$Consensus)

#create consensus peaksets from sample type consensuses for Responsive and Resistant sample groups
tamoxifen <- dba.peakset(tamoxifen,peaks=tamoxifen$masks$Consensus,consensus=DBA_CONDITION)
dba.show(tamoxifen,mask=tamoxifen$masks$Consensus)
dba.plotVenn(tamoxifen,17:18)

#retrieve the consensus peakset as GRanges 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=DBA$config$th, bUsePval=DBA$config$bUsePval,
            bNormalized=TRUE, attribute=DBA_GROUP, mask,
            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_DESEQ2
- DBA_DESEQ2_BLOCK
- DBA_EDGER
- 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 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, if available. Possible values:

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

mask — logical mask of samples to include when no groups are present.

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$AnalysisMethod,
                th=DBA$config$th, bUsePval=DBA$config$bUsePval,
                report, score, bLog=TRUE, mask, sites, sortFun=sd,
                correlations=TRUE, olPlot=DBA_COR,
               ColAttributes,RowAttributes, colSideCols, rowSideCols=colSideCols,
                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
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). Only significantly differentially bound sites will be used (subject to the th and bUsePval parameters). If mask is unspecified, only the samples in the contrast will be included. See dba.show(DBA, bContrast=TRUE) to get contrast numbers. If missing, uses scores in the main binding matrix.
method	analysis method (used in conjunction with contrast): • DBA_DESEQ2 • DBA_DESEQ2_BLOCK • DBA_EDGER • 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 (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_NORMALIZED
- DBA_SCORE_READS
- DBA_SCORE_CONTROL_READS
- DBA_SCORE_READS_MINUS
- DBA_SCORE_READS_FOLD
- DBA_SCORE_RPKM
- DBA_SCORE_RPKM_FOLD
- DBA_SCORE_RPKM_MINUS

bLog — Logical indicating that log2 values should be used. Only applicable with read count scores (not peak scores).

mask — mask indicating a subset of peaksets to use when using global binding matrix scores. If a contrast is specified, these peaksets will be included, but only the significantly differentially bound sites (using th, bUsePval, and/or report) will be included.

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 not equal to FALSE, 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

ColAttributes — Attribute or vector of attributes to plot for column color bars. If missing, all attributes with two or more unique non-NA values will be plotted. (For correlation heatmaps, DBA_GROUP will be plotted in the column color bar by default when a contrast is specified). A value of NULL indicates that no column color bar should be drawn. Allowable attribute values include:

- DBA_GROUP
- DBA_TISSUE
- DBA_FACTOR

- DBA_CONDITION
- DBA_TREATMENT
- DBA_REPLICATE
- DBA_CALLER

RowAttributes	Attribute or vector of attributes for row color bars. Row color bars are only allowed for correlation heatmaps. Same values as for ColAttributes parameter. Default is to draw a row color bar only if a contrast is specified, in which case the plotted attribute is DBA_GROUP (if present).
rowSideCols	Vector of colors to use in row color bars. Uses default colors if missing. Can also be a list of color vectors.
colSideCols	Vector of colors to use in column color bars. Uses default colors if missing. Can also be a list of color vectors.
margin	margin size of plot
colScheme	Color scheme; see colorRampPalette
distMethod	distance method for clustering; see Dist
...	passed on to heatmap.2, 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_DESEQ2,
th=0.05, bUsePval=F, mask, 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.

if correlations is FALSE, the sites used in the heatmap are returned in a GRanges object, in the row order they appear (top to bottom). The metadata contains a column for each sample (also in the order they are appear in the clustering plot), with the values being the actual plotted values.

Author(s)

Rory Stark

See Also

dba.overlap

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
sites <- dba.plotHeatmap(tamoxifen,contrast=1,th=1,
                         correlations=FALSE,sortFun=var)
sites

data(tamoxifen_counts)
#Examples of  heatmaps using DB sites with different subsets of samples
#exclude T47D
tamoxifen <- dba.contrast(tamoxifen,design=FALSE,
                          group1=tamoxifen$masks$Resistant,
                          group2=c(3:5,10:11))
tamoxifen <- dba.analyze(tamoxifen)

# regular heatmaps with samples from two contrast groups only
dba.plotHeatmap(tamoxifen, contrast=1)
#also include the T47D samples
dba.plotHeatmap(tamoxifen,contrast=1,mask=tamoxifen$masks$All)

#correlation heatmap without MCF7
plot(tamoxifen,contrast=1,mask=!tamoxifen$masks$MCF7)

# binding heatmap using only the MCF7 samples
dba.plotHeatmap(tamoxifen,contrast=1,mask=tamoxifen$masks$MCF7,correlations=FALSE)
```

---

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=DBA$config$th, bUsePval=DBA$config$bUsePval,
           fold=0, bNormalized=TRUE,
           factor="", bFlip=FALSE, bXY=FALSE, dotSize=.45,
           bSignificant=TRUE, highlight=NULL,
           bSmooth=TRUE, bLoess=TRUE, xrange, yrange, ...)
```

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=TRUE) to get contrast numbers. Alternatively, an MA plot can be generated without a specific contrast, plotting one set of samples against another. In this case, contrast should be a list on length one or two. Each element of the list should be either a logical sample mask, or a vector of sample numbers. If the second set of samples is jot specified (list is length one), all the samples other than those specified will be used for the second group. The list elements should be named; these names will be used as labels for the sample groups in the plot.
method	method or vector of methods to plot results for: • DBA_DESEQ2 • DBA_DESEQ2_BLOCK • DBA_EDGER • 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.
fold	will only include sites with fold change greater than this as significant (colored red). If fold is greater than zero, and an explicit design was used for the contrast, the p-value and FDR will be re-calculated based on testing for changes greater than the specified fold change. For a DESeq2 analysis, this involves including the fold when calling DESeq2::results. For a edgeR analysis, edgeR::glmTreat is used.
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.
bFlip	logical indicating that order of groups in contrast should be "flipped", allowing control of which sample group will have positive and which will have negative fold changes.

bXY	logical indicating whether to draw MA plot (FALSE) or XY scatter plot (TRUE).
dotSize	size of points on plot (cex).
bSignificant	Logical indicating if points corresponding to significantly differentially bound sites (based on contrast, th, bUsePval, and fold parameters) should be overlaid in red.
highlight	GRanges object with sites to highlight in green.
bSmooth	logical indicating that basic plot should be plotted using smoothScatter. Note that overlaid significant sites will be not plotted using a smoothing function.
bLoess	logical indicating that a MA plot should include a fitted loess curve.
xrange	vector of length 2 containing the desired minimum and maximum concentrations to plot.
yrange	vector of length 2 containing the desired minimum and maximum fold changes to plot.
...	passed to underlying plotting functions.

Author(s)

Rory Stark

See Also

dba.analyze

Examples

```
data(tamoxifen_analysis)

# default MA plot
dba.plotMA(tamoxifen)

# Show different normalizations
tamoxifen <- dba.normalize(tamoxifen,method=DBA_ALL_METHODS,
                           library=DBA_LIBSIZE_PEAKREADS, background=FALSE)
tamoxifen <- dba.analyze(tamoxifen, method=DBA_ALL_METHODS)

par(mfrow=c(3,2))
dba.plotMA(tamoxifen,th=0,bNormalized=FALSE,sub="NON-NORMALIZED")
dba.plotMA(tamoxifen,th=0,bNormalized=FALSE,sub="NON-NORMALIZED")

dba.plotMA(tamoxifen,method=DBA_DESEQ2,bNormalized=TRUE,
           sub="DESeq2_RLE-RiP")
dba.plotMA(tamoxifen,method=DBA_EDGER,bNormalized=TRUE,
           sub="edgeR_TMM-RiP")

tamoxifen <- dba.normalize(tamoxifen, method=DBA_ALL_METHODS,
                           normalize=DBA_NORM_LIB, background=FALSE)
tamoxifen <- dba.analyze(tamoxifen,method=DBA_ALL_METHODS)

dba.plotMA(tamoxifen,method=DBA_DESEQ2,bNormalized=TRUE,
```

```
           sub="DESeq2_LIB-FULL")
dba.plotMA(tamoxifen,method=DBA_EDGER,bNormalized=TRUE,
           sub="edgeR_LIB-FULL")

# MA plots of samples without a contrast
data(tamoxifen_counts)
par(mfrow=c(2,2))
dba.plotMA(tamoxifen,list(Resistant=tamoxifen$masks$Resistant,
                          Responsive=tamoxifen$masks$Responsive),
                          bNormalized=FALSE)
dba.plotMA(tamoxifen,list(MCF7=tamoxifen$masks$MCF7),
                          bNormalized=FALSE)
dba.plotMA(tamoxifen, list(Sample1=1), bNormalized=FALSE)
dba.plotMA(tamoxifen, list(Random=sample(1:11,5)), bNormalized=FALSE)

#XY plots (with raw and normalized data)
data(tamoxifen_analysis)
par(mfrow=c(1,2))
dba.plotMA(tamoxifen,bXY=TRUE,bSmooth=FALSE,bNormalized=FALSE,
           sub="NON_NORMALIZED")
dba.plotMA(tamoxifen,bXY=TRUE,bSmooth=FALSE,bNormalized=TRUE,
           sub="DESeq2-RLE-Background")
```

---

dba.plotPCA *PCA plot*

Description

Principal Component Analysis plot

Usage

```
dba.plotPCA(DBA, attributes, minval, maxval,
            contrast, method=DBA$config$AnalysisMethod,
            th=DBA$config$th, bUsePval=DBA$config$bUsePval,
            report, score, bLog=TRUE, mask, sites, label, cor=FALSE,
            b3D=FALSE, vColors, dotSize, labelSize, labelCols,
            components=1:3, ...)
```

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

Note that DBA_GROUP is a special attribute which will result in samples from each group in a contrast (if present) 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). If mask is unspecified, only the samples in the contrast will be included. 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_DESEQ2
- DBA_DESEQ2_BLOCK
- DBA_EDGER
- 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 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_NORMALIZED
- DBA_SCORE_CONTROL_READS
- DBA_SCORE_READS_MINUS
- DBA_SCORE_READS_FOLD
- DBA_SCORE_RPKM
- DBA_SCORE_RPKM_FOLD
- DBA_SCORE_RPKM_MINUS

bLog — Logical indicating that log2 values should be used. Only applicable to read count scores (not peak scores).

mask — mask indicating a subset of peaksets to use when using global binding matrix scores. If a contrast is specified, these peaksets will be included, but only the significantly differentially bound sites (using th, bUsePval, or report) will be included. See dba.mask.

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

label	A metadata field to use as a label in 2D plots. The value for this field will be written directly on the plot near the dot for each sample. Values can be any of those valid for the attributes parameter.
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.
labelSize	Scaling factor for labels if present. Default is 0.8.
labelCols	Vector of colors to use for labels. Default is "black".
components	Number(s) of the components to plot. Can be a vector of two or three component numbers, or a single integer. If an integer, that component, in addition to the succeeding one (b3D=FALSE) or two (b3D=TRUE) will be plotted.
...	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

trellis plot from lattice package; see xyplot

Note

uses rgl package for 3D plots (if available)

Author(s)

Rory Stark

See Also

dba.analyze, dba.plotHeatmap

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
p <- dba.plotPCA(tamoxifen,contrast=1)
p <- dba.plotPCA(tamoxifen,contrast=1,attributes=DBA_TISSUE)
p <- dba.plotPCA(tamoxifen,contrast=1,attributes=DBA_TISSUE,label=DBA_CONDITION)
p <- dba.plotPCA(tamoxifen,contrast=1,attributes=DBA_CONDITION,label=DBA_TISSUE)
p <- dba.plotPCA(tamoxifen,contrast=1,attributes=c(DBA_TISSUE,DBA_CONDITION),
                label=DBA_REPLICATE)
```

`dba.plotProfile` *Generate profiles and make profile heatmaps*

Description

Generates profiles and makes heatmap plots.

Usage

```
dba.plotProfile(Object, samples, sites, scores, labels,
                normalize=TRUE, merge=DBA_REPLICATE,
                maxSites=1000, absScores=TRUE,
                doPlot=is(Object,"profileplyr"),
                ...)
```

Arguments

`Object`	Either a `DBA` object, or a `profileplyr-class` object.
`samples`	Sample mask. A vector of `logical` or `numeric` values indicating which samples to be included in the plot. Alternatively, `samples` can be specified as a `list` of sample masks to specify sample groups (using list element names if present). If absent, all samples will be included; if `sites` indicates that the results of an analysis should be used, the samples involved in the specified contrast will be included (if it is a two-way contrast); these samples will be merged into two sample groups representing the two sides of the contrast. Some groups of samples may be merged as indicated in the `merge` parameter.
`sites`	`sites` is used to specify which sites are to be used in the heatmaps. It can be specified in a number of ways: • `GRanges` object containing a set of genomic intervals (eg. as returned by `dba.report`)

- logical or numeric vector of length > 1 indicating which sites to include in the heatmaps. If logical, vector should be same length as number of consensus sites binding matrix.
- GRangesList containing a list of GRanges, each containing a set of genomic intervals. Each element of the list will be plotted in a separate heatmap as a group of sites. If the constituent GRanges elements are named, the names will be used as labels for the site groups.
- A numeric value indicating a contrast on which an analysis has been run. In this case, all of the differentially-bound sites will be included, divided into two groups: a group of Gain sites (Fold > 0) and a group of Loss sites (Fold < 0).
- A report-based DBA object, as generated by dba.report. Each set of peaks in the object will be included as a separate group of sites.

If sites is absent, and an analysis has been completed, the first contrast will be used (sites=1); otherwise, all sites (subject to the maxSites limit) will be included.

scores — character string corresponding to the name of a metadata column containing numerical scores used to sort the sites (within each group).

These can be any of the mcols name values when passing in sites as a GRanges or GRangesList object, or the metadata fields in a report-based DBA object. If the Object is of type profileplyr-class, it can be any of its mcols names for columns corresponding to numeric values.

If scores=NULL, the sites will be sorted by their mean counts across all the samples.

labels — Either a vector of sample label names (one for each sample in the plot), or a set of attributes to include (positive values) or exclude (negative values). Attributes include:

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

normalize — logical indicating if the window counts should be normalized using the normalization established by dba.normalize.

Can also be a vector of normalization factors, once for each sample. All counts for a sample will be divided by the normalization factor for that sample.

merge — Set of attributes to be used to determine which samples should be merged. All samples that share the same values for all other attributes except those specified will be merged by taking their mean count score (after normalizing, if specified), and included as a single sample column.

Can also be specified as a list of vectors of sample numbers (relative to their order in mask). The samples corresponding to the values in each vector will be merged into a single sample.

maxSites	Maximum number of sites to include in a heatmap group. The top-scoring sites will be retained.
absScores	If TRUE, the absolute values for the score values (specified by the scores parameter) will be used for sorting sites. Useful for fold changes. If score values are greater than zero, this has no effect.
doPlot	logical indicating if the heatmap should be plotted. If FALSE, the profiles are generated and returned but not plotted.
...	additional parameters passed on to profileplyr::generateEnrichedHeatmap.

Details

This function enables the computation of peakset profiles and the plotting of complex heatmaps. It serves as a front-end to enable experiments analyzed using DiffBind to more easily use the profiling and plotting functionality provided by the profileplyr package written by Tom Carroll and Doug Barrows.

Processing proceeds in two phases.

In the first phase, specific peaksets are extracted from a DiffBind DBA object, and profiles are calculated for these peaks for set of samples in the DiffBind experiment. Profiles are calculated by counting the number of overlapping reads in a series of bins upstream and downstream of each peak center.

In the second phase, the derived profiles are plotted in a series of complex heatmaps showing the relative intensity of overlapping peaks in each bin for each peak in each sample, along with summary plots showing the average profile across the sites for each sample.

Due to the computational cost of this function, it is advised that the calculation of profiles and the plotting of heatmaps be separated into two calls, so that the profiles do not need to be re-generated if something goes wrong in the plotting. By default, when a DBA object is passed in to generate profiles, plotting is turned off and a profileplyr object is returned. When dba.plotProfile is called with a profileplyr object, a plot is generated by default.

More detailed documentation is included in a markdown demonstration script included with the DiffBind package. This can be located as follows:

```
system.file('extra/plotProfileDemo.Rmd',package='DiffBind')
```

An HTML version of the demonstration notebook can be accessed online at https://content.cruk.cam.ac.uk/bioinformatics/software/DiffBind/plotProfileDemo.html

Value

silently returns a profileplyr-class object.

Note

This function is currently disabled in this release of DiffBind because the profileplyr package is not installable in the current Bioconductor release. Calling dba.plotProfile will print a message and return NULL invisibly. Profile plotting will be restored in a future release when profileplyr is available again or a replacement backend is wired in.

Author(s)

Rory Stark

References

Carroll T, Barrows D (2020). profileplyr: Visualization and annotation of read signal over genomic ranges with profileplyr. DOI: 10.18129/B9.bioc.profileplyr

See Also

- profileplyr::profileplyr-class
- profileplyr::BamBigwig_to_chipProfile
- profileplyr::generateEnrichedHeatmap
- profileplyr::profileplyr (Vignette)
- DBA.config

Examples

```
# See plotProfileDemo notebook:
# system.file('extra/plotProfileDemo.Rmd',package='DiffBind')

data(tamoxifen_analysis)

# default Profile plot
## Not run: dba.plotProfile(tamoxifen)
```

---

dba.plotVenn *Draw 2-way, 3-way, or 4-way Venn diagrams of overlaps*

Description

Draws 2-way, 3-way, or 4-way Venn diagrams of overlaps

Usage

```
dba.plotVenn(DBA, mask, overlaps, label1, label2, label3, label4, main, sub,
             contrast, method=DBA$config$AnalysisMethod,
             th=DBA$config$th, bUsePval=DBA$config$bUsePval,
             bDB=TRUE, bNotDB, bAll=TRUE, bGain=FALSE, bLoss=FALSE,
             labelAttributes, DataType=DBA$config$DataType)
```

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
label4	label for fourth peakset in diagram
main	main title for plot
sub	subtitle for plot
contrast	contrast number(s) to use for results-based plots. This can be a vector of contrast numbers. See dba.show(DBA, bContrast=T) to get contrast numbers.
method	if contrast is specified, include results from analyses using this method or methods: • DBA_DESEQ2 • DBA_DESEQ2_BLOCK • DBA_EDGER • DBA_EDGER_BLOCK • DBA_ALL_METHODS • DBA_ALL_BLOCK • DBA_ALL_METHODS_BLOCK
th	if contrast is specified, use this significance threshold; all sites with FDR (or p-values, see bUsePval) less than or equal to this value will be considered differentially bound (DB).
bUsePval	if contrast is specified, this logical indicates whether to use FDR (FALSE) or p-value (TRUE) for thresholding.
bDB	if contrast is specified, this logical indicates that peaksets should include Differentially Bound (DB) sites (respecting the th, bUsePval, and fold parameters).
bNotDB	if contrast is specified, this logical indicates that peaksets should include non-Differentially Bound (non-DB) sites (respecting the th, bUsePval, and fold parameters).
bAll	if contrast is specified, this logical indicates peaksets combining peaks with both positive and negative fold changes should be included.
bGain	if contrast is specified, this logical indicates that peaksets with only positive fold changes should be included.
bLoss	if contrast is specified, this logical indicates that peaksets with only negative fold changes should be included.

labelAttributes
: if labels are not specified, use these attributes to create default labels:

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

  Only specified attributes that differ between peaksets will be used for labels; the ones that have the same value for all peaksets will be used as the default subtitle.

DataType
: if bReturnPeaksets is set to TRUE, 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.

  Alternatively, this can be set to:

  - DBA_DATA_DBAOBJECT

  to return a results-based DBA object, if a contrast is specified (see dba.report).

Value

Either a list of peaksets is returned invisibly (as described in dba.overlap), or, if DataType=DBA_DATA_DBAOBJECT, a results-based DBA object.

Note

When working with results overlaps (a least one contrast is specified), and results-oriented DBA object is generated internally (as described in dba.report). In some cases, it may be better to generate the DBA object explicitly (using dba.report or setting bReturnPeaksets=TRUE and DataType=DBA_DATA_DBAOBJECT). This include the case where several plots are being made of the same results set, and it takes a long time to generate the results-based DBA object, as well as the case where there are more than four results peaksets and a mask needs to be specified. I

This function relies on vennPlot in the systemPipeR package, written by Thomas Girke.

Author(s)

Rory Stark

See Also

dba.analyze, dba.overlap, dba.report, dba.plotPCA, vennPlot

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")
par(mfrow=c(1,1))
dba.plotVenn(Responsive,Responsive$masks$Consensus)

#4-way overlap
data(tamoxifen_peaks)
tamoxifen <- dba.peakset(tamoxifen, consensus=DBA_TISSUE)
par(mfrow=c(1,1))
dba.plotVenn(tamoxifen,tamoxifen$masks$Consensus,
             main="Tissue consensus overlaps")

#Venns of differentially bound sites
data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen,design="~Tissue+Condition")
tamoxifen <- dba.analyze(tamoxifen,method=c(DBA_EDGER,DBA_DESEQ2))
dba.plotVenn(tamoxifen,contrast=1,method=DBA_ALL_METHODS,
             bAll=FALSE,bGain=TRUE,bLoss=TRUE)
par(mfrow=c(2,1))
dba.plotVenn(tamoxifen,contrast=1,method=DBA_ALL_METHODS,
             bAll=FALSE,bGain=TRUE,bLoss=FALSE)
dba.plotVenn(tamoxifen,contrast=1,method=DBA_ALL_METHODS,
             bAll=FALSE,bGain=FALSE,bLoss=TRUE)

data(tamoxifen_counts)
tamoxifen <- dba.contrast(tamoxifen,design=FALSE,block=DBA_TISSUE)
tamoxifen <- dba.contrast(tamoxifen,design="~Tissue + Condition",
                          contrast=c("Condition","Responsive","Resistant"))
tamoxifen <- dba.analyze(tamoxifen,method=DBA_ALL_METHODS)
dba.plotVenn(tamoxifen,contrast=1:2,method=c(DBA_DESEQ2,DBA_DESEQ2_BLOCK))
tamoxifen.db <- dba.report(tamoxifen,contrast=1:2,method=DBA_ALL_METHODS_BLOCK,
                           bDB=TRUE)
dba.plotVenn(tamoxifen.db,mask=1:2)
```

```
dba.plotVenn(tamoxifen.db,mask=3:6)
```

---

dba.plotVolcano *Generate volcano plots of differential binding analysis results*

Description

Generates volcano plots of differential binding analysis results.

Usage

```
dba.plotVolcano(DBA, contrast=1, method=DBA$config$AnalysisMethod,
                            th=DBA$config$th, bUsePval=DBA$config$bUsePval,
                            fold=0, factor="", bFlip=FALSE,
                            bLabels=FALSE, maxLabels=50, dotSize=1,
                            bReturnSites=TRUE)
```

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=TRUE) to get contrast numbers.
method	method or vector of methods to plot results for: • DBA_DESEQ2 • DBA_DESEQ2_BLOCK • DBA_EDGER • DBA_EDGER_BLOCK
th	significance threshold; 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.
fold	will only include sites with fold change greater than this as significant (colored red). If fold is greater than zero, and an explicit design was used for the contrast, the p-value and FDR will be re-calculated based on testing for changes greater than the specified fold change. For a DESeq2 analysis, this involves including the fold when calling DESeq2::results. For a edgeR analysis, edgeR::glmTreat is used.
factor	string to be prepended to plot main title; e.g. factor name.
bFlip	logical indicating that order of groups in contrast should be "flipped", allowing control of which sample group will have positive and which will have negative fold changes.

bLabels	logical indicating that labels should be drawn on the plot. The labels are the site numbers, the row index in the (silently) returned set of significant sites. The maximum number of sites can be specified using maxLabels.
maxLabels	The maximum number of labels to use in the plot. Ignored if bLabels=FALSE.
dotSize	size of points on plot.
bReturnSites	If TRUE, silently returns the differential sites. If FALSE, the ggplot object is silently returned.

Details

Makes a volcano plot.

Value

Silently returns wither a GRanges object of the sites highlighted in red or a ggplot object.

Author(s)

Rory Stark

See Also

dba.analyze, dba.plotMA

Examples

```
data(tamoxifen_analysis)

# default volcano plot
dba.plotVolcano(tamoxifen)

# only highlight significant sites with at least 3x Fold Change
sigSites <- dba.plotVolcano(tamoxifen, fold=log2(3))

# use labels to find outlier sites
sigSites <- dba.plotVolcano(tamoxifen, fold=log2(5), th=0.01, bLabels=TRUE)
sigSites
```

---

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, method=DBA$config$AnalysisMethod,
           th=DBA$config$th, bUsePval=DBA$config$bUsePval,
           fold=0, bNormalized=TRUE, bFlip=FALSE, precision,
           bCalled=FALSE, bCounts=FALSE, bCalledDetail=FALSE,
           bDB, bNotDB, bAll=TRUE, bGain=FALSE, bLoss=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. When generating a report-based DBA object, this can be a vector of contrast numbers. If missing, defaults to first contrast for reports, and all contrasts when generating a report-based DBA object. See dba.show(DBA, bContrast=T) to get contrast numbers.
method	method used for analysis: • DBA_DESEQ2 • DBA_DESEQ2_BLOCK • DBA_EDGER • DBA_EDGER_BLOCK When generating a report-based DBA object (see bDB and bNotDB parameters below), a vector of methods may be supplied, including the shortcuts • DBA_ALL_METHODS • DBA_ALL_BLOCK • DBA_ALL_METHODS_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 log Fold value greater than equal to this will be included in the report. This should be supplied as a log2() value. If fold is greater than zero, and an explicit design was used for the contrast, the p-value and FDR will be re-calculated based on testing for changes greater than the specified fold change. For a DESeq2 analysis, this involves including the fold when calling DESeq2::results. For a edgeR analysis, edgeR::glmTreat is used.
bNormalized	logical indicating that normalized data (using normalization factors computed by differential analysis method) should be reported. When bNormalized=TRUE, read counts are adjusted by the normalization factors for calculating concentration values. Fold changes are reported using the potentially shrunk values computed by the underlying analysis package.

When bNormalized=FALSE, raw count data is used as the basis for reporting log concentration values, and Fold changes are reported based on subtracting the log concentration of one sample group from the other.

Confidence statistics (p-value/FDR) are always reported as computed by the underlying analysis package, which incorporate normalization factors.

bFlip — logical indicating that order of groups in contrast should be "flipped", allowing control of which sample group will have positive and which will have negative fold changes.

precision — If present, alters the default precision for the Concentration, Fold, p-value, and FDR values in the returned report. A value of 0 indicates maximum precision. Otherwise, it should be a 2-value vector. The first value controls how many digits to the right of the decimal to include for concentration and fold values. These second value control how many digits to the right of the decimal to include for the p-value and FDRs. Default is precision=2:3, unless DataType=DBA_DATA_SUMMARIZED_EXPERIMENT, in which case the default is 0 (full precision).

bCalled — logical indicating that peak caller status should be included. 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. Note that this option is only available if the consensus peakset was calculated by dba.count; if a consensus peakset was passed in explicitly using the peaks parameter, original peak origins are lost.

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.

bDB — logical indicating that a report-based DBA object should be generated, and that it should include Differentially Bound (DB) sites (respecting the th, bUsePval, and fold parameters).

bNotDB — logical indicating that a report-based DBA object should be generated, and that it should include non-Differentially Bound (non-DB) sites (respecting the th, bUsePval, and fold parameters).

bAll — logical indicating that a report-based DBA object should be generated, and that it should include peaksets combining peaks with both positive and negative fold changes.

bGain — logical indicating that a report-based DBA object should be generated, and that it should include peaksets with only positive fold changes.

bLoss — logical indicating that a report-based DBA object should be generated, and that it should include peaksets with only negative fold changes.

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

If set to DBA_DATA_SUMMARIZED_EXPERIMENT, the result will be a SummarizedExperiment object, with all the count data and sample metadata for the experiment. The contrast statistics will be included as metadata columns in the rowData of the object.

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

Value

if neither bDB or bNotDB is set to TRUE, a report dataframe or GRanges object is returned, 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, if present. The SampleID 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, if present. The SampleID 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.

If bDB or bNotDB is set to TRUE, a special DBA object is returned, containing peaksets based on sites determined to be differentially bound (or not) as specified using the bDB, bNotDB, bGain, bLoss, and bAll parameters. In this DBA object, the Tissue value will specify the direction of the change (Gain for positive fold changes, Loss for negative fold changes, and All for any fold change). The Factor value specifies if the peaks are differentially bound (DB) or not (!DB). The Condition value specifies the analysis method (e.g. edgeR), and the Treatment value is blank for unblocked analyses and set to block for blocked analyses.

Author(s)

Rory Stark

See Also

dba.analyze, DBA.config.

Examples

```
data(tamoxifen_analysis)

#Retrieve DB sites with FDR <  0.05
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

#Retrieve report as a SummarizedObject
tamoxifen.sset <- dba.report(tamoxifen, DataType=DBA_DATA_SUMMARIZED_EXPERIMENT)
tamoxifen.sset

#Retrieve report-based DBA object
data(tamoxifen_analysis)
tamoxifen <- dba.analyze(tamoxifen,method=DBA_ALL_METHODS)
tamoxifen.DB <- dba.report(tamoxifen,method=c(DBA_EDGER,DBA_DESEQ2),
                           bDB=TRUE)
tamoxifen.DB
dba.plotVenn(tamoxifen.DB,1:2)
```

---

dba.save *save DBA object*

Description

Writes out DBA object

Usage

```
dba.save(DBA, file='DBA', dir='.', pre='dba_', ext='RData',
         bRemoveAnalysis=FALSE, bRemoveBackground=FALSE,
         bCompress=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
bRemoveAnalysis	if TRUE, will remove the global DESeq2 and/or edgeR analysis objects. The analysis results will be retained. If the analysis objects are required after re-loading, they will be automatically re-generated.
bRemoveBackground	if TRUE, will remove the global binned background counts used for normalization. Any normalization factors calculated using these counts will be retained. If the the normalization factors need to be re-re-calculated after re-loading, the binned background counts will be automatically re-generated.
bCompress	logical indicating saved DBA object should be compressed as much as possible.

Value

string containing full path and filename.

Author(s)

Rory Stark

See Also

dba.load, DBA.config.

Examples

```
## Not run:
data(tamoxifen_peaks)
savefile <- dba.save(tamoxifen,'tamoxifenPeaks')
savefile
rm(tamoxifen)
tamoxifen <- dba.load('tamoxifenPeaks')
unlink(savefile)

## End(Not run)
```

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, bDesign=FALSE,
         th=DBA$config$th)
```

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_TREATMENT • DBA_REPLICATE • DBA_CONSENSUS • DBA_CALLER • DBA_CONTROL • DBA_READS • DBA_INTERVALS • DBA_FRIP
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.
bDesign	logical indicating whether the model design should be returned, if present. bContrasts must be FALSE for this parameter to be used.
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.

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=TRUE, th)
```

MODE: Return design associated with a DBA object:

```
dba.show(DBA,bDesign=TRUE)
```

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 four 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
`DB.edgeR.block`	Number of significantly differentially bound sites identified for blocking analysis using edgeR
`DB.DESeq.block`	Number of significantly differentially bound sites identified for blocking analysis using DESeq

Author(s)

Rory Stark

See Also

`dba`, `dba.peakset`, `dba.contrast` `dba.analyze`, `DBA.config`.

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_analysis)
```

```
dba.show(tamoxifen,bContrasts=TRUE)

#alternatively:
data(tamoxifen_analysis)
tamoxifen
tamoxifen$config$th <- .01
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_READS
DBA_ALL_ATTRIBUTES

DBA_INTERVALS
DBA_FRIP

DBA_GROUP

DBA_OLAP_PEAKS
DBA_OLAP_ALL
DBA_OLAP_RATE

DBA_COR
DBA_OLAP
DBA_INALL

DBA_SCORE_READS
DBA_SCORE_NORMALIZED
DBA_SCORE_CONTROL_READS
DBA_SCORE_READS_MINUS
```

```
DBA_SCORE_READS_FULL
DBA_SCORE_READS_MINUS_FULL
DBA_SCORE_READS_EFFECTIVE
DBA_SCORE_READS_MINUS_EFFECTIVE
DBA_SCORE_READS_FOLD
DBA_SCORE_RPKM
DBA_SCORE_RPKM_FOLD
DBA_SCORE_RPKM_MINUS
DBA_SCORE_TMM_READS_FULL
DBA_SCORE_TMM_READS_EFFECTIVE
DBA_SCORE_TMM_MINUS_FULL
DBA_SCORE_TMM_MINUS_EFFECTIVE
DBA_SCORE_TMM_READS_FULL_CPM
DBA_SCORE_TMM_READS_EFFECTIVE_CPM
DBA_SCORE_TMM_MINUS_FULL_CPM
DBA_SCORE_TMM_MINUS_EFFECTIVE_CPM
DBA_SCORE_SUMMIT
DBA_SCORE_SUMMIT_ADJ
DBA_SCORE_SUMMIT_POS
DBA_SCORE_FOLD
DBA_SCORE_CONCENTRATION
DBA_SCORE_CONC_NUMERATOR
DBA_SCORE_CONC_DENOMINATOR
DBA_SCORE_PVAL
DBA_SCORE_FDR

DBA_READS_DEFAULT
DBA_READS_BAM
DBA_READS_BED

DBA_EDGER
DBA_DESEQ2
DBA_EDGER_BLOCK
DBA_DESEQ2_BLOCK
DBA_EDGER_GLM
DBA_ALL_METHODS
DBA_ALL_BLOCK
DBA_ALL_METHODS_BLOCK

DBA_DATA_FRAME
DBA_DATA_GRANGES
DBA_DATA_RANGEDDATA
DBA_DATA_SUMMARIZED_EXPERIMENT
DBA_DATA_DBAOBJECT

DBA_BLACKLIST_CE10
DBA_BLACKLIST_CE11
DBA_BLACKLIST_DM3
```

```
DBA_BLACKLIST_DM6
DBA_BLACKLIST_GRCH37
DBA_BLACKLIST_GRCH38
DBA_BLACKLIST_HG19
DBA_BLACKLIST_HG38
DBA_BLACKLIST_MM9
DBA_BLACKLIST_MM10
DBA_BLACKLIST
DBA_GREYLIST
DBA_BLACKLISTED_PEAKS

DBA_LIBSIZE_DEFAULT
DBA_LIBSIZE_FULL
DBA_LIBSIZE_PEAKREADS
DBA_LIBSIZE_BACKGROUND
DBA_LIBSIZE_USER
DBA_NORM_DEFAULT
DBA_NORM_NATIVE
DBA_NORM_LIB
DBA_NORM_TMM
DBA_NORM_RLE
DBA_NORM_SPIKEIN
DBA_NORM_USER
DBA_NORM_OFFSETS
DBA_NORM_OFFSETS_ADJUST
DBA_OFFSETS_LOESS
DBA_OFFSETS_USER
```

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_READS	Number of reads counted in BAM file.
DBA_ALL_ATTRIBUTES	DBA peakset metadata: all attributes that can be used in certain plot labels (cf dba.plotVenn), equivalent to c(DBA_ID, DBA_TISSUE, DBA_FACTOR, DBA_CONDITION,DBA_TREATMENT DBA_REPLICATE, DBA_CALLER)
DBA_INTERVALS	DBA peakset metadata: Number of intervals in peakset

DBA_FRIP DBA peakset metadata: Fraction of Reads in Peaks (number of reads in intervals divided by total number of reads in library)

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_COR When plotting a heatmap from an overlap record, use the correlation value.

DBA_OLAP When plotting a heatmap from an overlap record, use the percentage overlap value.

DBA_INALL When plotting a heatmap from an overlap record, use the number of peaks in common to both samples.

DBA_SCORE_READS
dba.count score is number of reads in ChIP

DBA_SCORE_CONTROL_READS
dba.count score is number of reads in Control

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_READS_FULL
dba.count score is normalized ChIP read counts, using Full Library size

DBA_SCORE_READS_MINUS_FULL
dba.count score is normalized ChIP read counts minus Control read counts, using Full Library size

DBA_SCORE_READS_EFFECTIVE
dba.count score is normalized ChIP read counts, using Effective Library size

DBA_SCORE_READS_MINUS_EFFECTIVE
dba.count score is normalized ChIP read counts minus Control read counts, using Effective Library size

DBA_SCORE_NORMALIZED
dba.count score is normalized reads

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_RPKM_MINUS
dba.count score is RPKM of ChIP minus 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_SCORE_TMM_READS_FULL_CPM
: dba.count score is TMM normalized (using edgeR), using ChIP read counts and Full Library size, reported in counts-per-million.

DBA_SCORE_TMM_READS_EFFECTIVE_CPM
: dba.count score is TMM normalized (using edgeR), using ChIP read counts and Effective Library size, reported in counts-per-million.

DBA_SCORE_TMM_MINUS_FULL_CPM
: dba.count score is TMM normalized (using edgeR), using ChIP read counts minus Control read counts and Full Library size, reported in counts-per-million.

DBA_SCORE_TMM_MINUS_EFFECTIVE_CPM
: dba.count score is TMM normalized (using edgeR), using ChIP read counts minus Control read counts and Effective Library size, reported in counts-per-million.

DBA_SCORE_SUMMIT
: dba.count score is summit height (highest pile-up).

DBA_SCORE_SUMMIT_ADJ
: dba.count score is summit height (highest pile-up), adjusted for library size.

DBA_SCORE_SUMMIT_POS
: dba.count score is summit location (position of highest pile-up).

DBA_SCORE_FOLD
: score for report-based DBA object is Log Fold Change.

DBA_SCORE_CONCENTRATION
: score for report-based DBA object is Log Mean Concentration.

DBA_SCORE_CONC_NUMERATOR
: score for report-based DBA object is Log Mean Concentration of numerator (first group in contrast).

DBA_SCORE_CONC_DENOMINATOR
: score for report-based DBA object isLog Mean Concentration of denominator (second group in contrast).

DBA_SCORE_PVAL
: score for report-based DBA object is p-value.

DBA_SCORE_FDR
: score for report-based DBA object is FDR.

DBA_READS_DEFAULT
: When counting read files, use the file extension to determine the file type.

DBA_READS_BAM
: When counting read files, assume the file type is BAM, regardless of the file extension.

DBA_READS_BED
: When counting read files, assume the file type is BED (or zipped BED), regardless of the file extension.

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

DBA_DESEQ2
: differential analysis method: DESeq2 (using a single-factor GLM)

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

`DBA_DESEQ2_BLOCK`
: differential analysis method: DESeq2 with blocking factors (GLM)

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

`DBA_ALL_METHODS`
: use both analysis methods: `c(DBA_EDGER, DBA_DESEQ2)`

`DBA_ALL_BLOCK`
: report on block results for both analysis methods: `c(DBA_EDGER_BLOCK, DBA_DESEQ2_BLOCK)`

`DBA_ALL_METHODS_BLOCK`
: report on block results for all analysis methods, both blocked and unblocked: `c(DBA_ALL_METHODS, DBA_ALL_BLOCK)`

`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).

`DBA_DATA_SUMMARIZED_EXPERIMENT`
: Return report as a `SummarizedExperiment`.

`DBA_DATA_DBAOBJECT`
: Return a result-based DBA object from `dba.plotVenn`.

`DBA_BLACKLIST_HG19`
: Homo sapiens 19 (chromosomes have "chr")

`DBA_BLACKLIST_HG38`
: Homo sapiens 38 (chromosomes have "chr")

`DBA_BLACKLIST_GRCH37`
: Homo sapiens 37 (chromosomes are numbers)

`DBA_BLACKLIST_GRCH38`
: Homo sapiens 38 (chromosomes are numbers)

`DBA_BLACKLIST_MM9`
: Mus musculus 9

`DBA_BLACKLIST_MM10`
: Mus musculus 10

`DBA_BLACKLIST_CE10`
: C. elegans 10

`DBA_BLACKLIST_CE11`
: C. elegans 11

`DBA_BLACKLIST_DM3`
: Drosophila melanogaster 3

`DBA_BLACKLIST_DM6`
: Drosophila melanogaster 6

`DBA_BLACKLIST`
: Retrieve blacklist

`DBA_GREYLIST`
: Retrieve greylist

DBA_BLACKLISTED_PEAKS
: Retrieve blacklisted peaks

DBA_LIBSIZE_DEFAULT
: Default library size (DBA_LIBSIZE_FULL if no background, and DBA_LIBSIZE_CHR if background present)

DBA_LIBSIZE_FULL
: Full library size (all reads in library)

DBA_LIBSIZE_PEAKREADS
: Library size is Reads in Peaks

DBA_LIBSIZE_BACKGROUND
: Library size is Reads in Background

DBA_LIBSIZE_USER
: User supplied library sizes

DBA_NORM_DEFAULT
: Default normalization method

DBA_NORM_NATIVE
: "Native"" normalization method (TMM for DBA_EDGER and RLE for DBA_DESEQ2)

DBA_NORM_LIB
: Normalize by library size only

DBA_NORM_TMM
: Normalize using TMM method (edgeR)

DBA_NORM_RLE
: Normalize using RLE method (DESeq2)

DBA_NORM_SPIKEIN
: Normalize based on spike-ins

DBA_NORM_USER
: User supplied normalization factors

DBA_NORM_OFFSETS
: Use offsets instead of normalization factors

DBA_NORM_OFFSETS_ADJUST
: Use offsets instead of normalization factors; adjust based on library size (DESeq)

DBA_OFFSETS_LOESS
: Compute offsets using loess fit

DBA_OFFSETS_USER
: Use offsetrs supplied by user

Note

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

Author(s)

Rory Stark

DiffBind3 *Differences between DiffBind 3.0 and earlier versions*

Description

Notes on the differences between DiffBind 3.0 and previous versions, and how run in a "backward compatible" manner.

Overview

Beginning with version 3.0, DiffBind introduces substantial updates and new features that may cause scripts written for earlier versions to function differently (or not at all), as well as altering the results. This page givens details on these changes, and how to approximate results computed with earlier version if desired.

The major change in version 3.0 is in how the data are modeled. In previous versions, a separate model was derived for each contrast, including data only for those samples present in the contrast. Model design options were implicit and limited to either a single factor, or a subset of two-factor "blocked" designs. Starting in version 3.0, the default mode is to include all the data in a single model, allowing for any allowable design formula and any set of allowable contrasts.

Another change starting from version 3.0 is in how normalization is done. There are more normalization options, and more explicit control over them. The default normalization options have also changed, so reproducing a pre-3.0 analysis requires that normalization parameters to be specified.

It is recommended that existing analyses be re-run with the current software. Existing scripts should execute (with the exception of two normalization parameters which have been moved from dba.analyze to the new interface function dba.normalize.)

See the DiffBind vignette for more information on processing and analyzing ChIP-seq (and ATAC-seq) experiments.

Changes to Defaults

- **blacklist** is applied by default, if available, using automatic detection of reference genome.
- **greylists** are generated from controls and applied by default.
- **minimum read counts** are now 0 instead of being rounded up to 1 (this is now controllable).
- **centering peaks around summits** is now done by default using 401-bp wide peaks (recommend to use ‘summits=100‘ for ATAC-seq).
- **read counting** is now performed by ‘summarizeOverlaps()‘ by default, with single-end/paired-end counting automatically detected.
- **filtering** is performed by default; consensus peaks where no peak has and RPKM value of at least 1 in any sample are filtered.
- **control read subtraction** is now turned off by default if a greylist is present
- **normalization** is based on full library sizes by default for both ‘edgeR‘ and ‘DESeq2‘analyses.
- **score** is set to normalized values by default.

Backward compatibility

Most existing DiffBind scripts and saved objects will run correctly using version 3.0, but there may be differences in the results.

This section describes how to approximate earlier results for existing scripts and objects.

Running with saved DBA objects:: If a DBA object was created with an earlier version of DiffBind, and saved using the dba.save function, and loaded using the dba.load function, all settings should be preserved, such that running the analysis anew will yield the same results.

In order to re-run the analysis using the post-version 3.0 settings, the original script should be used to re-create the DBA object.

Re-running DiffBind scripts:: By default, if you re-run a DiffBind script, it will use the new defaults from version 3.0 and beyond. In order to re-analyze an experiment in the pre-version 3.0 mode, a number of defaults need to be changed.

When calling dba.count, the following defaults are changed:

- summits: This parameter is now set by default. Setting summits=FALSE will preempt re-centering each peak interval around its point of highest pileup.
- filter: The new default for this parameter is 1 and is based on RPKM values; previously it was set to filter=0 and was based on read counts.
- minCount: This is a new parameter representing a minimum read count value. It now default to 0; to get the previous behavior, set minCount=1.

The easiest way to perform subsequent processing in a pre-version 3.0 manner is to set a configuration option:

```
DBA$config$design <- FALSE
```

This will result in the appropriate defaults being set for the new interface function, dba.normalize (which does not need to be invoked explicitly.) The pre-version 3.0 settings for dba.normalize parameters are as follows:

- normalize: DBA_NORM_DEFAULT
- library: DBA_LIBSIZE_FULL
- background: FALSE

Note that two parameters that used to be available when calling dba.analyze have been moved:

- bSubControl: now integrated into dba.count. FALSE by default (unless a greylist has been added using dba.blacklist).
- bFullLibrarySize: now integrated into dba.normalize as an option for the library parameter. library=DBA_LIBSIZE_FULL is equivalent to bFullLibrarySize=TRUE, and library=DBA_LIBSIZE_PEAKR is equivalent to bFullLibrarySize=FALSE.

Author(s)

Rory Stark

See Also

The DiffBind vignette has been updated to show how to analyze experiments using version 3.0.

Index

∗ **package**
DiffBind-package, 2

calcThreshold, *16*
colorRampPalette, *47*

dba, 3, *3*, *8*, *31*, *41*, *69*
DBA object methods, 8
DBA tamoxifen resistance dataset, 11
DBA-config, *5*
DBA-config *(*DBA object methods*)*, 8
DBA-object *(*DBA object methods*)*, 8
dba.analyze, *3*, *5*, *9*, 12, *20*, *23*, *25*, *33*, *45*, *49, 50*, *52, 53*, *59*, *61–63*, *66*, *68, 69*, *77, 78*
dba.blacklist, *3*, *11, 12*, 14, *14*, *24*, *78*
DBA.config, *7*, *14*, *17*, *20*, *27*, *41*, *57*, *66, 67*, *69*
DBA.config *(*DBA object methods*)*, 8
dba.contrast, *3*, *9*, *12–14*, 17, *33*, *68, 69*
dba.count, *3*, *12–14*, *16*, 22, *24*, *33*, *64*, *78*
dba.load, *3*, *9*, 26, *67*, *78*
dba.mask, *3, 4*, *18*, 27, *34*, *38*, *52*, *58*
dba.normalize, *3*, *12–14*, *23*, 29, *31*, *55*, *77, 78*
DBA.object *(*DBA object methods*)*, 8
dba.overlap, *3*, 34, *46*, *48*, *58, 59*
dba.peakset, *3*, *5*, *7*, *23*, *31*, 37, *39, 40*, *69*
dba.plotBox, *3*, 42
dba.plotHeatmap, *3*, *35*, *37*, 44, *53*
dba.plotMA, *3*, 48, *62*
dba.plotPCA, *3*, 51, *59*
dba.plotProfile, *3*, *10*, 54, *56*
dba.plotVenn, *3*, *37*, 57, *72*, *75*
dba.plotVolcano, *3*, 61
dba.report, *3*, *14*, *35*, *39*, *46*, *52*, *54, 55*, *59*, 62
dba.save, *3*, *9*, *27*, *33*, 66, *78*
dba.show, *3*, *7*, *10*, *29*, *35*, *38*, *45*, *49*, *52*, *58*, *61*, *63*, 68
DBA_ALL_ATTRIBUTES *(*DiffBind - DBA global constant variables*)*, 70
DBA_ALL_BLOCK, *58*, *63*
DBA_ALL_BLOCK *(*DiffBind - DBA global constant variables*)*, 70
DBA_ALL_METHODS, *13*, *29*, *58*, *63*
DBA_ALL_METHODS *(*DiffBind - DBA global constant variables*)*, 70
DBA_ALL_METHODS_BLOCK, *58*, *63*
DBA_ALL_METHODS_BLOCK *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_CE10 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_CE11 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_DM3 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_DM6 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_GRCH37 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_GRCH38 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_HG19 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_HG38 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_MM10 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLIST_MM9 *(*DiffBind - DBA global constant variables*)*, 70
DBA_BLACKLISTED_PEAKS *(*DiffBind - DBA global constant variables*)*, 70
DBA_CALLER, *35*, *43*, *45*, *47*, *52*, *55*, *59*, *68*
DBA_CALLER *(*DiffBind - DBA global constant variables*)*, 70
DBA_CONDITION, *35*, *43*, *45*, *47*, *52*, *55*, *59*, *68*

DBA_CONDITION (DiffBind - DBA global constant variables), 70
DBA_CONSENSUS, *35*, *43*, *45*, *52*, *55*, *59*, *68*
DBA_CONSENSUS (DiffBind - DBA global constant variables), 70
DBA_CONTROL, *68*
DBA_CONTROL (DiffBind - DBA global constant variables), 70
DBA_COR, *46*
DBA_COR (DiffBind - DBA global constant variables), 70
DBA_DATA_DBAOBJECT, *59*
DBA_DATA_DBAOBJECT (DiffBind - DBA global constant variables), 70
DBA_DATA_FRAME, *9*, *35*, *59*, *65*
DBA_DATA_FRAME (DiffBind - DBA global constant variables), 70
DBA_DATA_GRANGES, *9*, *35*, *59*, *65*
DBA_DATA_GRANGES (DiffBind - DBA global constant variables), 70
DBA_DATA_RANGEDDATA, *9*, *35*, *59*, *65*
DBA_DATA_RANGEDDATA (DiffBind - DBA global constant variables), 70
DBA_DATA_SUMMARIZED_EXPERIMENT, *9*, *65*
DBA_DATA_SUMMARIZED_EXPERIMENT (DiffBind - DBA global constant variables), 70
DBA_DESEQ2, *13*, *29*, *35*, *43*, *45*, *49*, *52*, *58*, *61*, *63*
DBA_DESEQ2 (DiffBind - DBA global constant variables), 70
DBA_DESEQ2_BLOCK, *35*, *43*, *45*, *49*, *52*, *58*, *61*, *63*
DBA_DESEQ2_BLOCK (DiffBind - DBA global constant variables), 70
DBA_EDGER, *13*, *29*, *35*, *43*, *45*, *49*, *52*, *58*, *61*, *63*
DBA_EDGER (DiffBind - DBA global constant variables), 70
DBA_EDGER_BLOCK, *35*, *43*, *45*, *49*, *52*, *58*, *61*, *63*
DBA_EDGER_BLOCK (DiffBind - DBA global constant variables), 70
DBA_EDGER_GLM (DiffBind - DBA global constant variables), 70
DBA_FACTOR, *35*, *43*, *45*, *46*, *51*, *55*, *59*, *68*
DBA_FACTOR (DiffBind - DBA global constant variables), 70
DBA_FRIP, *68*
DBA_FRIP (DiffBind - DBA global constant variables), 70
DBA_GREYLIST (DiffBind - DBA global constant variables), 70
DBA_GROUP, *43*, *46*, *51*
DBA_GROUP (DiffBind - DBA global constant variables), 70
DBA_ID, *35*, *43*, *45*, *51*, *55*, *59*, *68*
DBA_ID (DiffBind - DBA global constant variables), 70
DBA_INALL, *46*
DBA_INALL (DiffBind - DBA global constant variables), 70
DBA_INTERVALS, *68*
DBA_INTERVALS (DiffBind - DBA global constant variables), 70
DBA_LIBSIZE_BACKGROUND (DiffBind - DBA global constant variables), 70
DBA_LIBSIZE_DEFAULT (DiffBind - DBA global constant variables), 70
DBA_LIBSIZE_FULL (DiffBind - DBA global constant variables), 70
DBA_LIBSIZE_PEAKREADS (DiffBind - DBA global constant variables), 70
DBA_LIBSIZE_USER (DiffBind - DBA global constant variables), 70
DBA_NORM_DEFAULT (DiffBind - DBA global constant variables), 70
DBA_NORM_LIB (DiffBind - DBA global constant variables), 70
DBA_NORM_NATIVE (DiffBind - DBA global constant variables), 70
DBA_NORM_OFFSETS (DiffBind - DBA global constant variables), 70
DBA_NORM_OFFSETS_ADJUST (DiffBind - DBA global constant variables), 70
DBA_NORM_RLE (DiffBind - DBA global constant variables), 70
DBA_NORM_SPIKEIN (DiffBind - DBA global constant variables), 70
DBA_NORM_TMM (DiffBind - DBA global constant variables), 70
DBA_NORM_USER (DiffBind - DBA global constant variables), 70
DBA_OFFSETS_LOESS (DiffBind - DBA global constant variables), 70

DBA_OFFSETS_USER (DiffBind - DBA global constant variables), 70
DBA_OLAP, *46*
DBA_OLAP (DiffBind - DBA global constant variables), 70
DBA_OLAP_ALL, *35*
DBA_OLAP_ALL (DiffBind - DBA global constant variables), 70
DBA_OLAP_PEAKS, *35*
DBA_OLAP_PEAKS (DiffBind - DBA global constant variables), 70
DBA_OLAP_RATE, *35*
DBA_OLAP_RATE (DiffBind - DBA global constant variables), 70
DBA_READS, *68*
DBA_READS (DiffBind - DBA global constant variables), 70
DBA_READS_BAM (DiffBind - DBA global constant variables), 70
DBA_READS_BED (DiffBind - DBA global constant variables), 70
DBA_READS_DEFAULT (DiffBind - DBA global constant variables), 70
DBA_REPLICATE, *35*, *43*, *45*, *47*, *52*, *55*, *59*, *68*
DBA_REPLICATE (DiffBind - DBA global constant variables), 70
DBA_SCORE_CONC_DENOMINATOR (DiffBind - DBA global constant variables), 70
DBA_SCORE_CONC_NUMERATOR (DiffBind - DBA global constant variables), 70
DBA_SCORE_CONCENTRATION (DiffBind - DBA global constant variables), 70
DBA_SCORE_CONTROL_READS, *46*, *52*
DBA_SCORE_CONTROL_READS (DiffBind - DBA global constant variables), 70
DBA_SCORE_FDR (DiffBind - DBA global constant variables), 70
DBA_SCORE_FOLD (DiffBind - DBA global constant variables), 70
DBA_SCORE_NORMALIZED, *46*, *52*
DBA_SCORE_NORMALIZED (DiffBind - DBA global constant variables), 70
DBA_SCORE_PVAL (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS, *46*, *52*
DBA_SCORE_READS (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_EFFECTIVE (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_FOLD, *46*, *52*
DBA_SCORE_READS_FOLD (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_FULL (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_MINUS, *46*, *52*
DBA_SCORE_READS_MINUS (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_MINUS_EFFECTIVE (DiffBind - DBA global constant variables), 70
DBA_SCORE_READS_MINUS_FULL (DiffBind - DBA global constant variables), 70
DBA_SCORE_RPKM, *46*, *52*
DBA_SCORE_RPKM (DiffBind - DBA global constant variables), 70
DBA_SCORE_RPKM_FOLD, *46*, *52*
DBA_SCORE_RPKM_FOLD (DiffBind - DBA global constant variables), 70
DBA_SCORE_RPKM_MINUS, *46*, *52*
DBA_SCORE_RPKM_MINUS (DiffBind - DBA global constant variables), 70
DBA_SCORE_SUMMIT (DiffBind - DBA global constant variables), 70
DBA_SCORE_SUMMIT_ADJ (DiffBind - DBA global constant variables), 70
DBA_SCORE_SUMMIT_POS (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_MINUS_EFFECTIVE (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_MINUS_EFFECTIVE_CPM (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_MINUS_FULL (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_MINUS_FULL_CPM (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_READS_EFFECTIVE

(DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_READS_EFFECTIVE_CPM (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_READS_FULL (DiffBind - DBA global constant variables), 70
DBA_SCORE_TMM_READS_FULL_CPM (DiffBind - DBA global constant variables), 70
DBA_TISSUE, *35*, *43*, *45*, *46*, *51*, *55*, *59*, *68*
DBA_TISSUE (DiffBind - DBA global constant variables), 70
DBA_TREATMENT, *35*, *43*, *45*, *47*, *52*, *55*, *59*, *68*
DBA_TREATMENT (DiffBind - DBA global constant variables), 70
DESeqDataSet, *13*, *14*
DiffBind, *8*, *56*, *77*, *78*
DiffBind (DiffBind-package), 2
DiffBind - DBA global constant variables, 70
DiffBind-config (DBA object methods), 8
DiffBind-package, 2
DiffBind.config (DBA object methods), 8
DiffBind3, *18*, 77
Dist, *47*

GRanges, *15*, *16*, *31*, *38*, *41*, *47*, *50*, *54*, *55*, *65*
GRangesList, *16*, *55*
GreyList, *17*

heatmap.2, *47*

lattice, *53*

mcols, *55*
mean, *46*
median, *46*
min, *46*

plot, *53*
plot.DBA (DBA object methods), 8
print.DBA (DBA object methods), 8

sd, *46*
smoothScatter, *50*
SummarizedExperiment, *6*, *31*, *65*, *75*
summarizeOverlaps, *10*, *24*, *25*
summary.DBA (DBA object methods), 8

tamoxifen (DBA tamoxifen resistance dataset), 11
tamoxifen.greylist (DBA tamoxifen resistance dataset), 11
tamoxifen_analysis (DBA tamoxifen resistance dataset), 11
tamoxifen_counts (DBA tamoxifen resistance dataset), 11
tamoxifen_greylist (DBA tamoxifen resistance dataset), 11
tamoxifen_peaks (DBA tamoxifen resistance dataset), 11

vennPlot, *59*

xyplot, *53*