Package 'MMDiff2'

November 12, 2026

Description This package detects statistically significant differences between read enrichment profiles in different ChIP-Seq samples. To take advantage of shape differences it uses Kernel methods (Maximum Mean Discrepancy, MMD).

Type Package

Title Statistical Testing for ChIP-Seq data sets

Version 1.40.0

Depends R (>= 3.3), Rsamtools, Biobase,

biocViews ChIPSeq, DifferentialPeakCalling, Sequencing, Software

License Artistic-2.0

Imports GenomicRanges, locfit, BSgenome, Biostrings, shiny, ggplot2, RColorBrewer, graphics, grDevices, parallel, S4Vectors, methods

ImportFrom utils txtProgressBar setTxtProgressBar read.csv

Suggests MMDiffBamSubset, MotifDb, knitr, BiocStyle, BSgenome.Mmusculus.UCSC.mm9

VignetteBuilder knitr

RoxygenNote 5.0.1

Collate 'DBAmmd-Class.R' 'AllGenerics.R' 'DBAmmd-Accessors.R' 'DBAmmd-Showmethods.R' 'DBAmmd-internal.R' 'MMDiff2_data.R' 'compDists.R' 'compHists.R' 'compPvals.R' 'createPeakMatrix.R' 'estimateFragmentCenters.R' 'getPeakReads.R' 'helperFcts.R' 'plotDISTS4Peak.R' 'plotDists.R' 'plotPeak.R' 'reportResults.R' 'runShinyMMDiff2.R' 'server.R' 'ui.R'

git_url https://git.bioconductor.org/packages/MMDiff2

git_branch RELEASE_3_23

git_last_commit 6c8c790

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Gabriele Schweikert [cre, aut], David Kuo [aut]

Maintainer Gabriele Schweikert <gschweik@staffmail.ed.ac.uk>

Contents

Cfp1-Peaks . . . 2
compDists . . . 3
compHists . . . 4
compPvals . . . 6
DBAmmd-Accessors . . . 7
DBAmmd-class . . . 9
estimateFragmentCenters . . . 11
getPeakReads . . . 12
metaData . . . 13
mm9-Genes . . . 14
MMD . . . 15
plotDists . . . 16
plotDISTS4Peak . . . 17
plotPeak . . . 17
reportResults . . . 19
runShinyMMDiff2 . . . 19
server.MMDiff2 . . . 20
ui.MMDiff2 . . . 21

Index **22**

`Cfp1-Peaks` *Peaks for Cfp1-data set*

Description

Subset of MACS called Peaks for Cfp-1 data set. Consensus Peaks were created using diffBind (see below).

Usage

```
data('Cfp1-Peaks')
```

Format

contains Peaks, a GRanges object with 500 ranges and 3 metadata columns

References

data taken from Clouaire et al., Genes and Development, 2012.

Examples

```
# data was created as follows:
## Not run:
library('MMDiffBamSubset')
dataDir <- system.file("extdata", package="MMDiffBamSubset")
library('DiffBind')
olddir <- setwd(dataDir)
DBA <- dba(sampleSheet="Cfp1.csv", minOverlap=3)
Peaks <- dba.peakset(DBA, bRetrieve = TRUE)
DBA <- dba.count(DBA, minOverlap=3)
setwd(olddir)
peaks <- dba.peakset(DBA, bRetrieve=TRUE)
C <- Counts(MMD)
idx <- which(C[,1]>150 & C[,3]>150&width(Peaks)>1000&width(Peaks)<5000)
Peaks <- Peaks[idx[1:500]]

## End(Not run)
```

`compDists` *Compute distances between Peaks*

Description

This function computes pairwise distances between histograms according to the dist.method (MMD, KS). For large data sets it is a bit time consuming.

Usage

```
compDists(MD, dist.method = "MMD", sigma = NULL, run.parallel = TRUE)
```

Arguments

`MD`	DBAmmd Object. This Object can be created using `DBAmmd()`.
`dist.method`	specify method used for distances between samples. Currently only Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) implemented. (DEFAULT: 'MMD')
`sigma`	sigma parameter of the RBF kernel, determining the distance (along the genome) at which fragment counts decorrelate. If set to NULL, 100 random Peaks are used to determine sigma heuristically according to the method described in the MMDiff paper [1]. (DEFUALT: NULL)
`run.parallel`	whether to run in parallel (currently no parallelization implemented) (DEFAULT: FALSE)

Value

DBAmmd object with updated slot `Dists`

Author(s)

Gabriele Schweikert <G.Schweikert@ed.ac.uk>

References

[1] Schweikert et al. BMC Genomics 2013 ...

See Also

DBAmmd, plotDists, plotDISTS4Peak, compPvals

Examples

```
## Example using a small data set provided with this package:

data("MMD")
MMD.1 <- compDists(MMD)

# To inspect the computed distances:
D <- Dists(MMD.1,dist.method='MMD')
head(D)

# To analyse the result:
plotDists(MMD.1)
```

---

compHists *Compute Peak histograms*

Description

This function computes histograms at pre-defined regions (peaks) from mapped fragments, i.e. fragment counts at genomic position. Note, in contrast to genomic coverage or density maps, this function uses a single position per fragment (usually its center) rather than the whole extend of the fragment. This results in a significant increase in resolution. The parameter whichPos determines whether fragment centers, start or end positions should be considered ('Center','Left','Right'). Results are stored as a list in the Hists slot of the DBAmmd Object, with one entry per peak. For each peak i, a (n x L_i) matrix is generated, where n is the number of samples and L_i is the number of bins used to cover the extend of the peak. Note, L_i varies between peaks of different lengths.

Usage

```
compHists(MD, bin.length = 20, whichPos = "Center")
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
bin.length	size of binning window (in bp) (DEFAULT: 20)
whichPos	specifies which relative positions of mapped fragments should to be considered. Can be one of: 'Left.p', 'Right.p', 'Right.p' and 'Left.n': Start and end positions of fragments mapping to positive or negative strand, respectively ('Right.p' and 'Left.n' are not available for single-end reads). Additionally inferred positions: 'Center.n','Center.p','Center','Left','Right'. (DEFAULT: 'Center')

Value

DBAmmd object with updated slot Hists

See Also

DBAmmd, getPeakReads, estimateFragmentCenters, plotPeak,

Examples

```
## Example using a small data set provided with this package:

data("MMD")
bin.length <- 20
MMD.1 <- compHists(MMD,bin.length)

# use \code{plotPeak()} to plot indivdual peaks:
Peak.id <- '6'
plotPeak(MMD.1, Peak.id=Peak.id)

# or explicitly using the histograms:
H <- Hists(MMD.1, whichPos='Center')
Sample <- 'WT.AB2'
Peak.idx <- match(Peak.id, names(Regions(MMD.1)))
plot(H[[Peak.idx]][Sample,],t='l')

# add peak cooridnates:
Peak <- Regions(MMD.1)[Peak.idx]
meta <- metaData(MMD.1)
PeakBoundary <- meta$AnaData$PeakBoundary
x.coords <- as.integer(colnames(H[[Peak.idx]])) + start(Peak) - PeakBoundary
plot(x.coords,H[[Peak.idx]]['WT.AB2',],t='l',
    xlab=names(H)[Peak.idx], ylab='counts', main=Sample)
```

compPvals *compute p-values*

Description

This function determines peak-specific p-values based on distances between sample histograms.

Usage

```
compPvals(MD, dist.method = "MMD", diff.method = "MMD.locfit")
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
dist.method	specify method used for distances between samples. Currently only Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) implemented. (DEFAULT: 'MMD')
diff.method	method used to determine p-values and false discovery rates. Currently only 'MMD.locfit' implemented. (DEFAULT: 'MMD.locfit')

Value

DBAmmd object with updated Contrasts slot.

See Also

DBAmmd,reportResults, plotDists,compDists

Examples

```
## Example using a small data set provided with this package:

data("MMD")
MMD.1 <- setContrast(MMD,contrast='byCondition')
MMD.1 <- compPvals(MMD.1)
reportResults(MMD.1)
```

DBAmmd-Accessors *Extract data from DBAmmd objects*

Description

This help file describes different ways to access the slots and values contained in a DBAmmd-class objects.

Usage

```
## S4 method for signature 'DBAmmd'
Genome(x)

## S4 method for signature 'DBAmmd'
Samples(x)

## S4 method for signature 'DBAmmd'
numPeaks(x)

## S4 method for signature 'DBAmmd'
numSamples(x)

## S4 method for signature 'DBAmmd'
metaData(x)

## S4 method for signature 'DBAmmd'
Regions(x)

## S4 method for signature 'DBAmmd'
Reads(x, whichPos = "Center")

## S4 method for signature 'DBAmmd'
Counts(x, whichCounts = "T")

## S4 method for signature 'DBAmmd'
Hists(x, whichPos = "Center")

## S4 method for signature 'DBAmmd'
Dists(x, dist.method = NULL)

## S4 method for signature 'DBAmmd'
Contrast(x, whichContrast = 1)

## S4 method for signature 'DBAmmd'
setRegions(x, Regions)

## S4 method for signature 'DBAmmd'
```

```
setContrast(x, contrast)
```

Arguments

x	a DBAmmd Object. An empty instance can be created using DBAmmd(). (See DBAmmd-class for more details.)
whichPos	specifies which relative positions of mapped fragments should to be considered. Can be one of: 'Left.p', 'Right.p', 'Right.p' and 'Left.n': Start and end positions of fragments mapping to positive or negative strand, respectively ('Right.p' and 'Left.n' are not available for single-end reads). Additionally inferred positions: 'Center.n','Center.p','Center','Left','Right'. (DEFAULT: 'Center')
whichCounts	can be 'T': total counts, or 'p','n': counts of reads mapping to positive, negative strand, respectively.
dist.method	specify method used for distances between samples. Currently only Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) implemented. (DEFAULT: 'MMD')
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)
Regions	GRanges Object specifying the Regions of Interesst / Peaks.
contrast	determines how to set a new contrast for differential analysis. A contrast can be automatically created either 'byCondition', or 'byTissue'. The Contrast can also be manually set (see vignette for details).

Value

`Genome(x)` returns the name of the used genome version, if set in the metaData.

`Samples(x)` returns the information which was provided in the SampleSheet.csv to describe the data.

`numPeaks(x)` returns the number of Peaks / Regions of Interest that are associated with the DBAmmd object.

`numSamples(x)` returns the number of samples associated with the DBAmmd object.

`metaData(x)` returns the metaData associated with the DBAmmd object.

`Regions(x)` returns the Peaks / Regions of Interest that are associated with the DBAmmd object.

`Reads(x,whichPos)` returns the Reads mapping to the Regions of Interest.

`Counts(x,whichCounts)` returns a m x n matrix containing the Counts of Reads mapping to the Peaks / Regions of Interest. Depending on the value of 'whichCounts', total counts ('T'), or counts of reads mapping to positive ('p'), or negative strand ('n') are returnt. See `getPeakReads` for more details.

`Hists(x,whichPos)` returns a list of matrices of length m (number of Peaks). Each matrix is a n x L_i matrix, where n is the number of samples and L_i is the number of bins used to cover the extend of the peak. Note, L_i varies between peaks of different lengths. See `compHists` for more details.

`Dists(x,dist.method)` returns a matrix containing distances between pairs of samples for each peak. See `compDists` for more details.

`Contrast(x,whichContrast)` returns the specified contrast.

`setRegions(x,Regions)` returns a DBAmmd Object with set Peaks / Regions of Interests.

`setContrast(x,contrast)` returns a DBAmmd Object with a set contrast.

See Also

DBAmmd-class

Examples

```
data("MMD")

Samples(MMD)
Genome(MMD)
numPeaks(MMD)
numSamples(MMD)
metaData(MMD)
R <- Regions(MMD)
Pos <- Reads(MMD)
C <- Counts(MMD)
H <- Hists(MMD)
D <- Dists(MMD)
C1 <- Contrast(MMD)
```

DBAmmd-class *Class DBAmmd*

Description

The DBAmmd Class defines a container for differential binding analysis using MMDiff2. For this class a number of methods is foreseen, among which accessors for every slot. As MetaData, it needs to contain the path to the data directory and the name of a sampleSheet csv file.

Value

DBAmmd Object

Constructor

DBAmmd()returns an empty DBAmmd Object.
DBAmmd(MetaData) initializes a DBAmmd Object for a new Experiment.
(See below and the package vignette for more details.)

Slots

MetaData: List containing an ExpData and an AnaData compartment. "ExpData" needs a dataDir and a SampleSheet entry. A genome entry, which should be a valid BSGenome name, is useful to find sequence motifs. (Note the genome version needs to correspond to the one used for the read alignment. Use available.genomes() to find the right name.) The AnaData entry is used to store and access parameters for the MMDiff2 Analysis, like the sigma of the RBF Kernel.

rowRanges: GRanges object containing Regions of Interests (Peaks)

`Reads`: List containing positions of mapped reads, i.e. exact start and end positions of mapped fragments. In the case of single-end reads, the left most postions of fragments mapping to the positive strands and the right most positions of fragments mapping to the negative strands are stored in "Left.p" and "Right.n". Use `getPeakReads` to fill this slot and `estimateFragmentCenters` to add the (estimated) positions of fragment centers.

`RawTotalCounts`: m x n matrix containing total counts of reads mapping to m peaks in n samples (including input samples)

`RawCounts.p`: m x n matrix containing counts of reads mapping to positive (forward) strand

`RawCounts.n`: m x n matrix containing counts of reads mapping to negative (reverse) strand

`Hists`: List of lists, each of length m (number of Peaks). Compartments could be 'Left.p','Right.n','Left.n','Right.p','Center. 'Center.p','Center','Left','Right', defining whether left or right ends or centers of fragments should be considered for positive ('p') or negative ('n') strand, or both strands combined. For a given compartment there is one entry per peak, which is a n x L_i matrix, where n is the number of samples and L_i is the number of bins used to cover the extend of the peak. Note, L_i varies between peaks of different lengths. See `compHists()` for more details.

`DISTs`: List with compartments for different methods to compute distances (e.g. MMD). Each compartment contains a m x N matrix with computed distances for each Peak between N pairs of samples. See `compDists()` for more details.

`mCounts`: (for internal use only)

`Contrasts`: List of lists. Each entry contains a contrast i.e. the definition of two groups that should be compared to each other in a differential analysis. A Contrast needs entries "name1", "name2" for group names, as well as group memberships given in "group1" and "group2". Results of a differential test for this contrast are stored in an entry given by the method name, e.g. "MMD.locfit"

Author(s)

Gabriele Schweikert

See Also

DBAmmd-Accessors,getPeakReads

Examples

```
## Example using a small data set provided in the MMDiffBamSubset package

# setting the Experiment meta data:
ExpData <- list(dataDir=system.file("extdata", package="MMDiffBamSubset"),
           sampleSheet="Cfp1.csv")

MetaData <- list('ExpData' = ExpData)

# Creating a DBAmmd data set:
MMD <- DBAmmd(MetaData)
```

estimateFragmentCenters *estimate center of fragments*

Description

This function computes average shifts between forward and reverse strand and applies it to estimate fragment centers.

Usage

```
estimateFragmentCenters(MD, shift = NULL, draw.on = FALSE)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
shift	can be set if the offset between forward and reverse strand is known (e.g. 1/2 median fragment size). In this case shift will not be estimated (DEFAULT: NULL)
draw.on	plot scatterplots for counts on forward vs reverse strand and histograms of determined shifts (DEFAULT: FALSE)

Value

DBAmmd object with updated slots Reads and MetaData.

See Also

DBAmmd, getPeakReads, compHists

Examples

```
## Example using a small data set provided with this package

data("MMD")
MMD.1 <- estimateFragmentCenters(MMD)

# To access centers of fragments:
Reads.C <- Reads(MMD.1,'Center')

# To access the determined shifts for each sample:
meta <- metaData(MMD.1)
meta$AnaData$Shifts
```

getPeakReads *Get reads from indexed bam files for defined regions*

Description

This function collects all short reads from bam files that map to pre-defined regions of interest. Note, that it fetches the exact start and end positions of mapped fragments, not the coverage. In the case of single-end reads, the left most postions of fragments mapping to the positive strands and the right most positions of fragments mapping to the negative strands are stored. To find centers of fragments use `estimateFragmentCenters()`. Positions are given relative to the start of the peak. Also computed are TotalCounts, i.e. number of fragments mapping to a peak region, as well as number of fragments mapping to forward and reverse strand.

Usage

```
getPeakReads(MD, PeakBoundary = 200, pairedEnd = FALSE,
  run.parallel = FALSE)
```

Arguments

`MD`	DBAmmd Object. This Object can be created using `DBAmmd()`.
`PeakBoundary`	Defines flanking regions of peaks. The estimated fragment length is a good guess (DEFAULT: 200).
`pairedEnd`	whether the reads are paired-end (paired-end is currently not fully tested) (DEFAULT: FALSE)
`run.parallel`	whether to run in parallel (currently no parallelization implemented) (DEFAULT: FALSE)

Value

DBAmmd object with updated slots

See Also

DBAmmd, estimateFragmentCenters

Examples

```
## Example using a small data set provided in the MMDiffBamSubset package

# setting the Experiment meta data:
ExpData <- list(dataDir=system.file("extdata", package="MMDiffBamSubset"),
           sampleSheet="Cfp1.csv")

MetaData <- list('ExpData' = ExpData)

# Creating a DBAmmd data set:
```

```
MMD <- DBAmmd(MetaData)

# defining a small Region for which to get reads:
Regions <- GRanges(seqnames=c('chr1'),
           IRanges(start = c(4560912,4677889), end = c(4562680,4679681)))
MMD <- setRegions(MMD,Regions)
MMD <- getPeakReads(MMD)

# To access Left ends of fragments mapping to positive strand:
Reads.L <- Reads(MMD,'Left.p')

# To access Right ends of fragments mapping to negative strand:
Reads.R <- Reads(MMD,'Right.n')

# To access Matrix of TotalCounts:
C.t <- Counts(MMD,whichCounts='T')

# Counts on positive strand:
C.p <- Counts(MMD,whichCounts='p')

# Counts on negative strand:
C.n <- Counts(MMD,whichCounts='n')
```

---

metaData *Generics for DBAmmd-Class*

Description

Generics for DBAmmd-Class

Usage

```
metaData(x, ...)

Regions(x, ...)

Reads(x, ...)

Counts(x, ...)

Hists(x, ...)

Dists(x, ...)

Contrast(x, ...)

numPeaks(x, ...)
```

```
numSamples(x, ...)

Samples(x, ...)

Genome(x, ...)

setRegions(x, ...)

setContrast(x, ...)
```

Arguments

x	a DBAmmd Object. An empty instance can be created using `DBAmmd()`. (See `DBAmmd-class` for more details.)
...	additional parameters

`mm9-Genes` *mm9-Genes*

Description

Subset of Genes from the mm9 annotation that overlap with example Peaks in the Cfp1-Peaks file.

Usage

```
data('mm9-Genes')
```

Format

contains, GR a GRanges object with 800 ranges

Examples

```
# data was created as follows:
## Not run:
data('Cfp1-Peaks')
library(TxDb.Mmusculus.UCSC.mm9.knownGene)
txdb <- TxDb.Mmusculus.UCSC.mm9.knownGene #shorthand (for convenience) txdb
GR <- transcripts(txdb)
ov <- findOverlaps(GR,Peaks)
GR <- GR[queryHits(ov)]
save(file = 'data/mm9-Genes.rData',GR)

## End(Not run)
```

MMD *DBAmmd Object for Cfp1 example*

Description

DBAmmd Object for Cfp1 example

Usage

```
data('MMD')
```

Examples

```
# data was created as follows:
## Not run:
library('MMDiff2')
library('MMDiffBamSubset')
# create metaData:
ExperimentData <- list(genome='BSgenome.Mmusculus.UCSC.mm9',
                       dataDir=system.file("extdata", package="MMDiffBamSubset"),
                       sampleSheet="Cfp1.csv")
MetaData <- list('ExpData' = ExperimentData)
MMD <- DBAmmd(MetaData)
data("Cfp1-Peaks")

MMD <- setRegions(MMD,Peaks)
MMD <- getPeakReads(MMD,pairedEnd=FALSE, run.parallel=FALSE)

MMD <- DBAmmd(MetaData)
MMD <- setRegions(MMD,Peaks)
MMD <- getPeakReads(MMD,pairedEnd=FALSE, run.parallel=FALSE)
MMD <- estimateFragmentCenters(MMD, shift=NULL, draw.on=FALSE)
MMD <- compHists(MMD, bin.length=20)
MMD <- compDists(MMD, dist.method = "MMD", run.parallel = FALSE)
group1 <- Samples(MMD)$Condition==1
names(group1) <- Samples(MMD)$SampleID
group2 <- Samples(MMD)$Condition==2
names(group2) <-  Samples(MMD)$SampleID
con <- list(group1=group1,
           group2=group2,
           name1='WT-Resc',
           name2='KO')

MMD <- compPvals(MMD, contrasts=list(con))

## End(Not run)
```

plotDists *plotDists*

Description

scatterplot showing distances between peaks

Usage

```
plotDists(MD, dist.method = "MMD", whichContrast = 1, which.group1 = NULL,
  which.group2 = NULL, diff.method = "MMD.locfit", bUsePval = FALSE,
  th = 0.1, title = NULL, what = 3, xlim = NULL, ylim = NULL,
  xlog10 = TRUE, Peak.IDs = NULL, withLegend = TRUE,
  shiny_df_opt = FALSE)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
dist.method	specify method used for distances between samples. Currently only Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) implemented. (DEFAULT: 'MMD')
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)
which.group1	subset samples from group1 (DEFAULT: NULL)
which.group2	subset samples from group2 (DEFAULT: NULL)
diff.method	which method to use to determine significant peaks (DEFAULT: 'MMD.locfit')
bUsePval	if TRUE p-values instead of FDRs are used (DEFAULT: FALSE)
th	significance threshold for differential called peaks (DEFAULT: 0.1)
title	an overall title for the plot (DEFAULT: NULL)
what	which dists to overlay: 1: only between group distances, 2: between and within group distances, 3: between and within group distances, and significant peaks highlightend (DEFAULT: 3)
xlim	specify x range (DEFAULT: NULL)
ylim	specify y range (DEFAULT: NULL)
xlog10	should x range be plotted in log10 scale (DEFAULT: TRUE)
Peak.IDs	Highlight specific subset of peaks (DEFAULT: NULL)
withLegend	(DEFAULT: TRUE)
shiny_df_opt	Option returns a dataframe for shiny (DEFAULT: FALSE)

Examples

```
data("MMD")
plotDists(MMD, whichContrast=1)
```

plotDISTS4Peak *plotDISTS4Peak*

Description

showing all distances for one region

Usage

```
plotDISTS4Peak(MD, Peak.id, dist.method = "MMD", whichContrast = 1,
  Zoom = TRUE, xlim = NULL, ylim = NULL, xlog10 = TRUE, title = NULL)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
Peak.id	Peak id to specify which Peak to plot. (coresponding to names of Regions(MD))
dist.method	specify method used for distances between samples. Currently only Maximum Mean Discrepancy (MMD) and Kolmogorov-Smirnov (KS) implemented. (DEFAULT: 'MMD')
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)
Zoom	(DEFAULT: TRUE)
xlim	specify x range (DEFAULT: NULL)
ylim	specify y range (DEFAULT: NULL)
xlog10	should x range be plotted in log10 scale (DEFAULT: TRUE)
title	(DEFAULT: NULL)

Examples

```
dev.off()
load(system.file("data/MMD.RData", package="MMDiff2"))
plotDISTS4Peak(MMD,Peak.id = '6',dist.method='MMD', whichContrast=1)
```

plotPeak *plot Peak*

Description

This function plots histograms of fragment positions over a pre defined regions of interests / peaks. Can also show occurences of Sequence motifs and annotated objects (e.g. genes).

Usage

```
plotPeak(MD, Peak.id, Sample.ids = NULL, NormMethod = NULL,
  plot.input = FALSE, whichPos = "Center", whichContrast = NULL,
  Motifs = NULL, Motifcutoff = "80%", anno = NULL, xaxt = NULL,
  xlim = NULL, ylim = NULL)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
Peak.id	Peak id to specify which Peak to plot. (coresponding to names of Regions(MD))
Sample.ids	which samples to draw. If NULL all samples are drawn. (DEFAULT: NULL)
NormMethod	whether to apply normailzation factors. currently no normalization method implemented (DEFAULT: None)
plot.input	whether to plot input controls (DEFAULT: TRUE)
whichPos	specifies which relative positions of mapped fragments should to be considered. Can be one of: 'Left.p', 'Right.p', 'Right.p' and 'Left.n': Start and end positions of fragments mapping to positive or negative strand, respectively ('Right.p' and 'Left.n' are not available for single-end reads). Additionally inferred positions: 'Center.n','Center.p','Center','Left','Right'. (DEFAULT: 'Center')
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)
Motifs	TF binding sites (DEFAULT: NULL)
Motifcutoff	(Default: "80%")
anno	either a GRanges objects containing annotated objects, e.g. genes, or a list of GRanges Objects. (Default: NULL)
xaxt	(Default: NULL)
xlim	(Default: NULL)
ylim	(Default: NULL)

Examples

```
dev.off()
data("MMD")
plotPeak(MMD,Peak.id='6',plot.input=FALSE)

# add annotation (Overlapping genes)
data("mm9-Genes")
GR <- list(UCSCKnownGenes = GR)
plotPeak(MMD, Peak.id='6', plot.input = FALSE, anno=GR)

# add TF binding sites
library('MotifDb')
motifs <- query(query(MotifDb, 'Mmusculus'), 'E2F')
plotPeak(MMD, Peak.id='6', plot.input = FALSE,
       Motifs=motifs,Motifcutoff="80%")

# split peaks by contrast
```

```
plotPeak(MMD, Peak.id='6', plot.input = FALSE, whichContrast=1,
       Motifs=motifs,Motifcutoff="80%",anno=GR)
```

`reportResults` *report results*

Description

retrieve results of differential binding analysis

Usage

```
reportResults(MD, diff.method = "MMD.locfit", th = 0.1, whichContrast = 1,
  rm.oulier = TRUE, bUsePval = FALSE)
```

Arguments

`MD`	DBAmmd Object. This Object can be created using `DBAmmd()`.
`diff.method`	method used to determine p-values and false discovery rates. Currently only 'MMD.locfit' implemented. (DEFAULT: 'MMD.locfit')
`th`	significance threshold for differential called peaks (DEFAULT: 0.1)
`whichContrast`	index determining which of the set contrast should be used. (DEFAULT: 1)
`rm.oulier`	if TRUE, significant peaks with high within-group distances are not reported. (DEFAULT: TRUE)
`bUsePval`	if TRUE p-values instead of FDRs are used (DEFAULT: FALSE)

Examples

```
data("MMD")
res <- reportResults(MMD)
```

`runShinyMMDiff2` *Shiny Application for interactive visualization of MMD,GMD and Pearson Difference as well as plotting peaks*

Description

Shiny Application for interactive visualization of MMD,GMD and Pearson Difference as well as plotting peaks

Usage

```
runShinyMMDiff2(MD, whichContrast = 1)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)

Examples

```
 if(interactive()){
  data("MMD")
runShinyMMDiff2(MMD)
}
```

---

server.MMDiff2 *Shiny server code for interactive visualization of MMD distances, peak plots, and MMD distances by sample.*

Description

Shiny server code for interactive visualization of MMD distances, peak plots, and MMD distances by sample.

Usage

```
server.MMDiff2(MD, whichContrast = 1)
```

Arguments

MD	DBAmmd Object. This Object can be created using DBAmmd().
whichContrast	index determining which of the set contrast should be used. (DEFAULT: 1)

Examples

```
 if(interactive()){
    data("MMD")
    runShinyMMDiff2(MMD)
}
```

`ui.MMDiff2` *ui component for interactive visualization of MMD,GMD and Pearson Difference as well as plotting peaks*

Description

ui component for interactive visualization of MMD,GMD and Pearson Difference as well as plotting peaks

Usage

```
ui.MMDiff2(MD)
```

Arguments

`MD`	DBAmmd object

Examples

```
 if(interactive()){
load(system.file("data/MMD.RData", package="MMDiff2"))
runShinyMMDiff2(MMD)
}
```

Index

∗ **data**
  Cfp1-Peaks, 2
  mm9-Genes, 14
  MMD, 15

Cfp1-Peaks, 2
compDists, 3, *6*, *8*
compHists, 4, *8*, *11*
compPvals, *4*, 6
Contrast *(*metaData*)*, 13
Contrast,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
Counts *(*metaData*)*, 13
Counts,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7

DBAmmd, *4–6*, *11, 12*
DBAmmd *(*DBAmmd-class*)*, 9
DBAmmd-Accessors, 7
DBAmmd-class, 9
Dists *(*metaData*)*, 13
Dists,DBAmmd-method *(*DBAmmd-Accessors*)*,
  7

estimateFragmentCenters, *5*, 11, *12*

Genome *(*metaData*)*, 13
Genome,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
getPeakReads, *5*, *8*, *10, 11*, 12

Hists *(*metaData*)*, 13
Hists,DBAmmd-method *(*DBAmmd-Accessors*)*,
  7

metaData, 13
metaData,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
mm9-Genes, 14
MMD, 15

numPeaks *(*metaData*)*, 13
numPeaks,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
numSamples *(*metaData*)*, 13
numSamples,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7

plotDists, *4*, *6*, 16
plotDISTS4Peak, *4*, 17
plotPeak, *5*, 17

Reads *(*metaData*)*, 13
Reads,DBAmmd-method *(*DBAmmd-Accessors*)*,
  7
Regions *(*metaData*)*, 13
Regions,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
reportResults, *6*, 19
runShinyMMDiff2, 19

Samples *(*metaData*)*, 13
Samples,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
server.MMDiff2, 20
setContrast *(*metaData*)*, 13
setContrast,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7
setRegions *(*metaData*)*, 13
setRegions,DBAmmd-method
  *(*DBAmmd-Accessors*)*, 7

ui.MMDiff2, 21