Package ‘epihet’

April 15, 2020

Title Determining Epigenetic Heterogeneity from Bisulfite Sequencing Data

Version 1.2.0

Description epihet is an R-package that calculates the epigenetic heterogeneity between cancer cells and/or normal cells. The functions establish a pipeline that take in bisulfite sequencing data from multiple samples and use the data to track similarities and differences in epipolymorphism,proportion of discordantly methylated sequencing reads (PDR),and Shannon entropy values at epialleles that are shared between the samples.epihet can be used to perform analysis on the data by creating pheatmaps, box plots, PCA plots, and t-SNE plots. MA plots can also be created by calculating the differential heterogeneity of the samples. And we construct co-epihet network and perform network analysis.

Depends R(>= 3.6), GenomicRanges, IRanges, S4Vectors, ggplot2, foreach, Rtsne, igraph

License Artistic-2.0

biocViews DNAMethylation, Epigenetics, MethylSeq, Sequencing, Software

Imports data.table, doParallel, EntropyExplorer, graphics, stats, grDevices, pheatmap, utils, qvalue, WGCNA, ReactomePA

Encoding UTF-8

RoxygenNote 6.0.1

Suggests knitr, clusterProfiler, ggfortify, org.Hs.eg.db, rmarkdown

VignetteBuilder knitr

BuildVignettes TRUE

URL `https://github.com/TheJacksonLaboratory/epihet`

BugReports `https://github.com/TheJacksonLaboratory/epihet/issues`

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

git_branch RELEASE_3_10

git_last_commit b51b4f4

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Xiaowen Chen [aut, cre],
Haitham Ashoor [aut],
Ryan Musich [aut],
Mingsheng Zhang [aut],
Jiahui Wang [aut],
Sheng Li [aut]

Maintainer Xiaowen Chen <Xiaowen.Chen@jax.org>

R topics documented:

background *example data background*

Description

background: A data frame containing 31995 elements as background used for pathway enrichment analysis

datTraits: Clinical traits containing OS,EFS,age

DEG: Differentially expressed genes compared CEBPA-sil vs.normal

DEH: DEH loci

diffhetmatrix: A differentially heterogeneity matrix

moduledm: Module information for CEBPA-dm mutation samples

modulesil: Module information for CEBPA-sil mutation samples

promoter: The promoter region annotation file

sharedmatrix: Epipolymorphism values for 6 patients on DEH loci

Usage

```
data(background)
background

data(datTraits)
datTraits

data(DEG)
DEG

data(DEH)
DEH

data(diffhetmatrix)
diffhetmatrix

data(moduledm)
moduledm

data(modulesil)
modulesil

data(promoter)
promoter

data(sharedmatrix)
sharedmatrix
```

Format

background: A data frame with 31995 rows and 1 variables:

gene background gene list

Value

A data frame

A large GRanges object

---

compMatrix *Make Comparison Matrix*

Description

A matrix is created for pca/hclust/tsne which contains read number, average methylation levels, pdr, epipolymorphism, and Shannon entropy values across multiple samples at the same loci using read number in a GenomicRanges object

Usage

```
compMatrix(epi.gr, outprefix = NULL, readNumber = 60, p = 1,
  cores = 5, sve = FALSE)
```

Arguments

`epi.gr` An input file containing the read number, locus, pdr, epipolymorphism, and Shannon entropy values stored in a list of GenomicRanges objects

`outprefix` The prefix name of the outputted matrix file. ’sve’ must be set to TRUE (default: NULL)

`readNumber` The lowest number of reads required for each loci (default: 60)

`p` Percentage (as decimal) of matching samples required to determine a match at a given locus, e.g. a value of 0.75 requires 75% of the samples to have an epiallele at a common loci in order to add the loci to the matrix (default: 1)

`cores` The number of cores to be used for parallel execution (default: 5)

`sve` A boolean to save the comparison matrix (default: FALSE)

Value

A large matrix containing values (pdr, etc.) at the same loci

Examples

```
p1.GR<-GRanges(seqnames = Rle(c(”chr22”), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c(”-”, ”+”, ”+”, ”+”, ”-”))),
values.loci = c(”327:350:361:364”,”821:837:844:849”,
”838:845:850:858”,”755:761:771:773”,”761:771:773:781”),
values.read1 = c(92,72,68,176,176),values.meth1=c(84,93,94,96,95),
values.shannon=c(0.4,0.5,0.5,0.2,0.5),values.pdr=c(0.6,0.25,0.23,0.15,0.17),
values.epipoly=c(0.48,0.42,0.38,0.27,0.3))

p2.GR<-GRanges(seqnames = Rle(c(”chr22”), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c(”-”, ”+”, ”+”, ”+”, ”-”))),
values.loci = c(”327:350:361:364”,”821:837:844:849”,
”838:845:850:858”,”755:761:771:773”,”761:771:773:781”),
values.read1 = c(107,102,102,76,76),values.meth1=c(88,66,69,71,94),
values.shannon=c(0.12,0.25,0.54,0.23,0.25),
values.pdr=c(0.38,1,0.97,1,0.13),
values.epipoly=c(0.57,0.42,0.28,0.18,0.23))

N1.GR<-GRanges(seqnames = Rle(c(”chr22”), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c(”-”, ”+”, ”+”, ”+”, ”-”))),
values.loci = c(”327:350:361:364”,”821:837:844:849”,
”838:845:850:858”,”755:761:771:773”,”761:771:773:781”),
values.read1 = c(112,112,112,68,76),values.meth1=c(82,60,91,71,90),
values.shannon=c(0.15,0.26,0.34,0.24,0.15),
values.pdr=c(0.32,0.57,0.37,0.37,0.13),
values.epipoly=c(0.57,0.42,0.28,0.38,0.23))

N2.GR<-GRanges(seqnames = Rle(c(”chr22”), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c(”-”, ”+”, ”+”, ”+”, ”-”))),
values.loci = c(”327:350:361:364”,”821:837:844:849”,
”838:845:850:858”,”755:761:771:773”,”761:771:773:781”),
values.read1 = c(385,78,70,96,96),values.meth1=c(96,81,87,87,93),
values.pdr=c(0.15,0.52,0.48,0.25,0.25),
values.epipoly=c(0.26,0.58,0.58,0.37,0.37),
values.shannon=c(0.12,0.25,0.54,0.23,0.25))

GR.List<-list(p1=p1.GR,p2=p2.GR,N1=N1.GR,N2=N2.GR)
comp.Matrix <- compMatrix(epi.gr = GR.List, outprefix = NULL,
readNumber = 60, p = 1, cores = 1, sve = FALSE)
```

diffHet *Calculate Differential Heterogeneity*

Description

From a user-inputted value and two subtype groups, calculates the mean values for both subtypes at each loci. The heterogeneity difference is calculated and the p-values and adjusted p-values are calculated if the heterogeneity difference is greater than a given cutoff

Usage

```
diffHet(compare.matrix, value, group1, group2, subtype,
  het.dif.cutoff = 0.2, permutations = 1000, p.adjust.method = "fdr",
  cores = 5)
```

Arguments

compare.matrix	The comparison matrix generated from the compMatrix() function
value	The value to be used in calculations. Possible values are ’read’, ’pdr’, ’meth’, ’epipoly’, and ’shannon’
group1	The first subtype group to be compared
group2	The second subtype group to be compared
subtype	A dataframe containing the subtype information for the samples in the comparison matrix. The row names should be the names of the samples and there should be one column containing the subtype information for each sample.
het.dif.cutoff	A number representing the cutoff for the heterogeneity difference. If the heterogeneity difference is greater than the cutoff value, than the p-value and adjusted p-value are calculated for the loci. If the heterogeneity difference is less than the cutoff value, than the p-value and adjusted p-value are set to NA. (default: 0.20)
permutations	The number of permutations for the EntropyExplorer function. Value must be set to ’shannon’. (default: 1000)
p.adjust.method	The method to be used as a parameter in p.adjust() function. Possible methods are ’holm’, ’hochberg’, ’hommel’, ’bonferroni’, ’BH’, ’BY’, ’fdr’, and ’none’.(default: ’fdr’)
cores	The number of cores to be used for parallel execution. Not available for ’shannon’ values. (default: 5)

Value

A dataframe containing chromosome number, loci, mean of group1, mean of group2, heterogeneity difference, and the p-value and adjusted p-value for the loci with a heterogeneity difference greater than the cutoff

epiBox *Make Boxplot from Comparison Matrix*

Description

From a user-inputted value, finds the mean of that value for each sample, then creates a boxplot comparing the values for each subtype.

Usage

```
epiBox(compare.matrix, value, type, box.colors = NULL,
  add.points = FALSE, points.colors = NULL, pdf.height = 10,
  pdf.width = 10, sve = FALSE)
```

Arguments

`compare.matrix` The comparison matrix generated from the compMatrix() function

`value` The value to be graphed in the boxplot. Possible values are 'read', 'pdr', 'meth', 'epipoly', and 'shannon'

`type` A dataframe containing the type information for the samples in the comparison matrix. The row names should be the names of the samples and there should be one column containing the type information for each sample.

`box.colors` A vector of colors to be used as the fill color for each boxplot. If not entered, the default colors of ggplot are used. (default: NULL)

`add.points` A boolean stating if the individual points for each sample mean should be displayed over the box plots (default: FALSE)

`points.colors` A vector of colors to be used as the color of the individual points for each sample mean. One color is used per subtype. (default: NULL)

`pdf.height` An integer representing the height (in inches) of the outputted boxplot pdf file (default: 10)

`pdf.width` An integer representing the width (in inches) of the outputted boxplot pdf file (default: 10)

`sve` A boolean to save the plot (default: FALSE)

Value

a data frame containing the mean epigenetic heterogeneity for each sample

Examples

```
comp.Matrix<-data.frame(
p1=c(0.6,0.3,0.5,0.5,0.5,0.6,0.45,0.57,0.45,0.63,0.58,0.67,0.5,0.42,0.67),
p2=c(0.62,0.63,0.55,0.75,0.84,0.58,1,0.33,1,0.97,0.57,0.68,0.73,0.72,0.82),
p3=c(0.72,0.53,0.62,0.69,0.37,0.85,1,0.63,0.87,0.87,0.82,0.81,0.79,
0.62,0.68),
N1=c(0.15,0.24,0.15,0.26,0.34,0.32,0.23,0.14,0.26,0.32,0.12,0.16,0.31,
0.24,0.32),
N2=c(0.32,0.26,0.16,0.36,0.25,0.37,0.12,0.16,0.41,0.47,0.13,0.52,0.42,
0.41,0.23),
N3=c(0.21,0.16,0.32,0.16,0.36,0.27,0.24,0.26,0.11,0.27,0.39,0.5,0.4,
0.31,0.33),
type=rep(c("pdr","epipoly","shannon"),c(5,5,5)),
location=rep(c("chr22-327:350:361:364","chr22-755:761:771:773",
"chr22-761:771:773:781","chr22-821:837:844:849","chr22-838:845:850:858"),
3),stringsAsFactors =FALSE )

subtype <- data.frame(Type= c(rep('CEBPA_sil', 3), rep('Normal', 3)),
row.names <- colnames(comp.Matrix)[1:6],stringsAsFactors = FALSE)

epiBox(compare.matrix = comp.Matrix, value = 'epipoly',
type <- subtype, box.colors = NULL, add.points = FALSE,
points.colors <- NULL, pdf.height = 10, pdf.width = 10,
sve = FALSE)
```

epiMA *Make MA Plot*

Description

Creates an MA plot from the differential heterogeneity data calculated from the diffHet() function. For each loci, graphs the average of both group means on the x-axis and the heterogeneity difference on the y-axis. Graphs coordinates with significant adjusted p-values in red.

Usage

```
epiMA(pval.matrix, padjust.cutoff = 0.05, pch = ".", sve = FALSE)
```

Arguments

pval.matrix	The data frame returned from the diffHet() function that contains means, p-values, adjusted p-values, and heterogeneity difference
padjust.cutoff	The adjusted p-value cutoff to confirm a significant value. (default: 0.05)
pch	The plotting character to be used in the MA plot (default: '.')
sve	A boolean to save the plot (default: FALSE)

Value

A figure

Examples

```
diff.het.matrix<-data.frame(chromosome=c(rep("1",10)),
loci=paste("loci",seq_len(10),sep="-"),subtype.mean=c(0.21,0.23,0.37,0.26,
0.29,0.31,0.29,0.13,0.12,0.093),Normal.mean=c(0.01,0.01,0.01,0.02,
0.02,0.01,0.01,0.79,0.73,0.79),het.dif=c(0.20,0.220,0.360,0.240,0.270,
0.300,0.280,-0.660,-0.610,-0.697),p.value=c(3.08e-03,1.43e-02,9.27e-03,
3.45e-02,2.99e-02,3.68e-02, 4.60e-02, 5.65e-10, 9.18e-10,
9.98e-11),p.adjust=c(8.84e-03,2.76e-02, 2.04e-02, 5.01e-02,
4.56e-02, 5.24e-02, 6.08e-02, 3.74e-08, 5.22e-08,
1.06e-08),type=rep("pdr",10))

epiMA(pval.matrix = diff.het.matrix, padjust.cutoff = 0.05,
pch = ".", sve = TRUE)
```

epiMap *Make Pheatmap from Comparison Matrix*

Description

Creates a pheatmap for the top 'loci.percent' of values of max standard deviation from the comparison matrix generated by compMatrix(). The rows represent the loci of the epiallele and the columns represent the sample names. The columns can be annotated by adding annotation information as a parameter.

Usage

```
epiMap(compare.matrix, value, annotate,
  clustering_distance_rows = "euclidean",
  clustering_distance_cols = "euclidean",
  clustering_method = "complete", annotate.colors = NA,
  color = colorRampPalette(c("blue", "white", "red"))(1000),
  loci.percent = 0.1, show.rows = FALSE, show.columns = FALSE,
  font.size = 6, pdf.height = 10, pdf.width = 10, sve = FALSE, ...)
```

Arguments

`compare.matrix` The comparison matrix generated from the compMatrix() function

`value` The value to be graphed in the pheatmap. Possible values are 'read', 'pdr', 'meth', 'epipoly', and 'shannon'.

`annotate` A dataframe containing the annotation information for the columns of the pheatmap. The row names must be the names of the samples. The columns (any number) are the annotations. E.g. a column called 'TET2' with factors 'Pos' and 'Neg' for each sample that is positive or negative for the TET2 gene

`clustering_distance_rows`
Distance measure used in clustering rows.

`clustering_distance_cols`
Distance measure used in clustering columns.

`clustering_method`
clustering method used.

`annotate.colors`
A list containing the colors for the annotation information. Each element in the list is a vector of colors with names that correspond to the columns of 'annotate'.

`color` a vector of colors used in heatmap.

`loci.percent` The top percentage of loci, as a decimal, to be displayed on the pheatmap based on standard deviation, e.g. a value of 0.20 is equivalent to the top 20% of loci (default: 0.10)

`show.rows` A boolean stating if the row names should be displayed on the pheatmap (default: FALSE)

`show.columns` A boolean stating if the column names should be displayed on the pheatmap (default: FALSE)

`font.size` An integer representing the font size to be used for the pheatmap labels (default: 6)

`pdf.height`	An integer representing the height (in inches) of the pdf file for the pheatmap (default: 10)
`pdf.width`	An integer representing the width (in inches) of the pdf file for the pheatmap (default: 10)
`sve`	A boolean to save the plot (default: FALSE)
`...`	any arguments in the function pheatmap()

Value

A pheatmap object that contains the tree data for both rows and columns and the final pheatmap plot

Examples

```
comp.Matrix<-data.frame(
p1=c(0.6,0.3,0.5,0.5,0.5,0.6,0.45,0.57,0.45,0.63,0.58,0.67,0.5,0.42,0.67),
p2=c(0.62,0.63,0.55,0.75,0.84,0.58,1,0.33,1,0.97,0.57,0.68,0.73,0.72,0.82),
p3=c(0.72,0.53,0.62,0.69,0.37,0.85,1,0.63,0.87,0.87,0.82,0.81,0.79,
0.62,0.68),
N1=c(0.15,0.24,0.15,0.26,0.34,0.32,0.23,0.14,0.26,0.32,0.12,0.16,0.31,
0.24,0.32),
N2=c(0.32,0.26,0.16,0.36,0.25,0.37,0.12,0.16,0.41,0.47,0.13,0.52,0.42,
0.41,0.23),
N3=c(0.21,0.16,0.32,0.16,0.36,0.27,0.24,0.26,0.11,0.27,0.39,0.5,0.4,
0.31,0.33),
type=rep(c("pdr","epipoly","shannon"),c(5,5,5)),
location=rep(c("chr22-327:350:361:364","chr22-755:761:771:773",
"chr22-761:771:773:781","chr22-821:837:844:849","chr22-838:845:850:858"),
3),stringsAsFactors =FALSE )

subtype <- data.frame(Type= c(rep('CEBPA_sil', 3), rep('Normal', 3)),
row.names = colnames(comp.Matrix)[1:6],stringsAsFactors = FALSE)

pmap <- epiMap(compare.matrix = comp.Matrix,
value = 'epipoly',annotate = subtype,
clustering_distance_rows = "euclidean",
clustering_distance_cols = "euclidean",
clustering_method = "complete",annotate.colors = NA,
color= colorRampPalette(c("blue","white","red"))(1000),
loci.percent = 1, show.rows = FALSE,
show.columns = TRUE, font.size = 15,
pdf.height = 10, pdf.width = 10, sve = TRUE)
```

`epiNetwork` *Construct co-epihet network based on DEH loci.*

Description

Construct co-epihet network for DEH loci or for genes with genome region containing DEH loci using WGCNA and identify modules that are significantly associated with the measured clinical traits for co-epihet DEH loci network,we identify genes with genome region containing DEH loci in each module.

Usage

```
epiNetwork(node.type = "locus", DEH, compare.matrix, value = "pdr",
  group, subtype, datTraits = NULL, annotation.obj,
  networktype = "signed", method = "pearson", prefix = NULL,
  mergeCutHeight = 0.25, minModuleSize = 30)
```

Arguments

`node.type`	a character suggest node type in network. Possible values are 'locus','gene' (default:locus)
`DEH`	the dataframe containing the chromosome number, loci and strand information of DEH loci generated from diffHet() function
`compare.matrix`	The comparison matrix generated from the compMatrix() function.
`value`	A character, which is used to identify DEH loci. Possible values are 'pdr', 'epipoly',and 'shannon'(default:pdr)
`group`	The subtype group to be used to construct network
`subtype`	A dataframe containing the subtype information for the samples in the comparison matrix. The row names should be the names of the samples and there should be one column containing the subtype information for each sample
`datTraits`	a dataframe containing the clinical traits for all patients from the subtype group
`annotation.obj`	a GRanges object containing gene annotation information
`networktype`	network type in WGCNA (default:signed)
`method`	character string specifying the correlation to be used in WGCNA (default:pearson)
`prefix`	character string containing the file name base for files containing the consensus topological overlaps in WGCNA
`mergeCutHeight`	a numeric, dendrogram cut height for module merging (default: 0.25)
`minModuleSize`	a numeric, minimum module size for module detection in WGCNA (default: 30)

Value

a list, if node type is CpG site, it contains the epigenetic heterogeneity matrix for patients module information, gene.list which is a data frame containing genes with genome region containing DEH loci from one module if node type is gene,it contains the epigenetic heterogeneity matrix for patients and module information.

`epiPathway` *pathway annotation*

Description

pathway identification significantly enriched by genes in one module.

Usage

```
epiPathway(gene.list, cutoff = 0.05, prefix = NA, pdf.height = 10,
  pdf.width = 10)
```

Arguments

`gene.list` a data frame generated from network.construct() function. The first column is gene entrez ID, the second column is module lable, the third column is module color

`cutoff` Cutoff value of pvalue for pathway enrichment (default:0.05)

`prefix` a prefix for PDF file name

`pdf.height` An integer representing the height (in inches) of the outputted boxplot pdf file (default: 10)

`pdf.width` An integer representing the width (in inches) of the outputted boxplot pdf file (default: 10)

Value

a data frame containing pathways that are significantly enriched by genes from one module

Examples

```
genelist<-data.frame(ENTREZID=c("2902","2905","3360","286223","59338",
"344018","5144","55001","7410","730051","55743","6804","200634","2802",
"2260","651","2104","23432","10505","23194","9855","7101",
"389136","124857","1829","3164","3754","8614","9469","3217","9578",
"10516","10630"),label=rep(18,33),color=rep("lightgreen",33),
stringsAsFactors = FALSE)
pathway <- epihet::epiPathway(genelist,cutoff = 0.05,
                              prefix="CEBPA_sil",pdf.height = 10,
                              pdf.width = 10)
```

`epiPCA` *Make PCA Plot from Comparison Matrix*

Description

From a user-inputted value, creates a PCA plot from the sample data and colors each point by the subtype information provided.

Usage

```
epiPCA(compare.matrix, value, type, points.colors = NULL,
  frames = FALSE, frames.colors = NULL, probability = FALSE,
  pdf.height = 10, pdf.width = 10, sve = FALSE)
```

Arguments

`compare.matrix` The comparison matrix generated from the compMatrix() function

`value` The value to be graphed in the PCA plot

`type` A dataframe containing the type information for the samples in the comparison matrix. The row names should be the names of the samples and there should be one column containing the type information for each sample.

`points.colors` A vector to be used as the color of the individual points for each sample. One color is used per type. the names of vector is the types(default: NULL) `frames` A boolean stating if the frames should be drawn around the points for each subtype cluster. (default: False)

`frames.colors` A vector of colors to be used as the color of the frames for each subtype cluster. (default: NULL)

`probability` A boolean stating if the frames should be drawn as probability ellipses around the points for each subtype cluster. Both ’probability’ and ’frames’ must be set to TRUE to have effect. (default: False)

`pdf.height` An integer representing the height (in inches) of the outputted PCA plot pdf file (default: 10)

`pdf.width` An integer representing the width (in inches) of the outputted PCA plot pdf file (default: 10)

`sve` A boolean to save the plot (default: FALSE)

Value

A PCA plot

Examples

```
library(ggfortify)
comp.Matrix<-data.frame(
p1 = c(0.6,0.3,0.5,0.5,0.5,0.6,0.45,0.57,0.45,0.63,0.58,0.67,0.5,0.42,0.67),
p2 = c(0.62,0.63,0.55,0.75,0.84,0.58,1,0.33,1,0.97,0.57,0.68,0.73,0.72,0.82),
p3 = c(0.72,0.53,0.62,0.69,0.37,0.85,1,0.63,0.87,0.87,0.82,0.81,0.79,
0.62,0.68),
N1=c(0.15,0.24,0.15,0.26,0.34,0.32,0.23,0.14,0.26,0.32,0.12,0.16,0.31,
0.24,0.32),
N2=c(0.32,0.26,0.16,0.36,0.25,0.37,0.12,0.16,0.41,0.47,0.13,0.52,0.42,
0.41,0.23),
N3=c(0.21,0.16,0.32,0.16,0.36,0.27,0.24,0.26,0.11,0.27,0.39,0.5,0.4,
0.31,0.33),
type=rep(c(”pdr”,”epipoly”,”shannon”),c(5,5,5)),
location=rep(c(”chr22-327:350:361:364”,”chr22-755:761:771:773”,
”chr22-761:771:773:781”,”chr22-821:837:844:849”,”chr22-838:845:850:858”),
3),stringsAsFactors =FALSE )

subtype <- data.frame(Type= c(rep('CEBPA_sil', 3), rep('Normal', 3)),
row.names = colnames(comp.Matrix)[1:6],stringsAsFactors = FALSE)

epiPCA(compare.matrix = comp.Matrix, value = 'epipoly',
              type = subtype, points.colors = NULL,
            frames = FALSE, frames.colors = NULL,
            probability = FALSE, pdf.height = 10,
            pdf.width = 10, sve = TRUE)
```

epiTSNE *Make TSNE Plot from Comparison Matrix*

Description

From a user-inputted value, creates a TSNE plot from the sample data and colors each point by the subtype information provided.

Usage

```
epiTSNE(compare.matrix, value, type, points.colors = NULL, theta = 0.5,
  curTheme = NULL, perplexity = 5, max_iter = 1000,
  pdf.height = 10, pdf.width = 10, sve = FALSE)
```

Arguments

`compare.matrix`	The comparison matrix generated from the compMatrix() function
`value`	The value to be graphed in the PCA plot
`type`	A dataframe containing the type information for the samples in the comparison matrix. The row names should be the names of the samples and there should be one column containing the type information for each sample.
`points.colors`	A vector of colors to be used as the color of the individual points for each sample. One color is used per subtype. (default: NULL)
`theta`	A decimal representing the theta parameter for the Rtsne() function. Represents the speed/accuracy trade-off (0.0 is exact TSNE) (default: 0.5)
`curTheme`	the theme of ggplot2 to control control the appearance of all non-data components of the plot
`perplexity`	An integer representing the perplexity parameter for the Rtsne() function (default: 30)
`max_iter`	An integer representing the max_iter parameter for the Rtsne() function. Represents the number of iterations (default: 1000)
`pdf.height`	An integer representing the height (in inches) of the outputted TSNE plot pdf file (default: 10)
`pdf.width`	An integer representing the width (in inches) of the outputted TSNE plot pdf file (default: 10)
`sve`	A boolean to save the plot (default: FALSE)

Value

A T-SNE plot

Examples

```
comp.Matrix<-data.frame(
p1=c(0.6,0.3,0.5,0.5,0.5,0.6,0.45,0.57,0.45,0.63,0.58,0.67,0.5,0.42,0.67),
p2=c(0.62,0.63,0.55,0.75,0.84,0.58,1,0.33,1,0.97,0.57,0.68,0.73,0.72,0.82),
p3=c(0.72,0.53,0.62,0.69,0.37,0.85,1,0.63,0.87,0.87,0.82,0.81,0.79,
0.62,0.68),
N1=c(0.15,0.24,0.15,0.26,0.34,0.32,0.23,0.14,0.26,0.32,0.12,0.16,0.31,
0.24,0.32),
N2=c(0.32,0.26,0.16,0.36,0.25,0.37,0.12,0.16,0.41,0.47,0.13,0.52,0.42,
0.41,0.23),
N3=c(0.21,0.16,0.32,0.16,0.36,0.27,0.24,0.26,0.11,0.27,0.39,0.5,0.4,
0.31,0.33),
type=rep(c("pdr","epipoly","shannon"),c(5,5,5)),
location=rep(c("chr22-327:350:361:364","chr22-755:761:771:773",
"chr22-761:771:773:781","chr22-821:837:844:849","chr22-838:845:850:858"),
3),stringsAsFactors =FALSE )

subtype <- data.frame(Type= c(rep('CEBPA_sil', 3), rep('Normal', 3)),
row.names = colnames(comp.Matrix)[1:6],stringsAsFactors = FALSE)

epiTSNE(compare.matrix = comp.Matrix, value = 'epipoly',
type = subtype, points.colors = NULL, theta = 0.5,
perplexity = 1, max_iter = 1000, pdf.height = 10,
pdf.width = 10, sve = TRUE)
```

jaccard *Jaccard score calculation between modules from two subtypes.*

Description

Jaccard score calculation based on the common genes in two modules from two subtypes.

Usage

```
jaccard(module.subtype1, module.subtype2)
```

Arguments

module.subtype1
: a data frame generated from the epiNetwork() function. The module information of subtype1, the first column is module nodes, the second column is module label, the third column is module color.

module.subtype2
: a data frame generated from the epiNetwork() function. The module information of subtype1, the first column is module nodes, the second column is module label, the third column is module color.

Value

A matrix containing Jaccard scores.

Examples

```
data(modulesil)
data(moduledm)
jaccard.matrix <- jaccard(modulesil, moduledm)
```

makeGR *Make List of GenomicRanges Object*

Description

Creates a GenomicRanges object for each methclone output file

Usage

```
makeGR(files, ids, cores = 5, sve = FALSE)
```

Arguments

`files`	A vector of input files containing methclone output files, the suffix of files should be methClone_out.gz
`ids`	A vector of sample ids for the files
`cores`	The number of cores to be used for parallel execution (default: 5)
`sve`	A boolean to save the GenomicRanges object (default: FALSE)

Value

A list, each element is a data frame of GenomicRanges objects containing pdr, epipolymorphism, and Shannon entropy values for each input file. Saves as an epi.gr.rda extension

Examples

```
path <- system.file('extdata', package = 'epihet')
files <- dir(path = path, pattern = 'methClone_out.gz',
             recursive = TRUE, full.names = TRUE)
ids <- basename(dirname(files))
GR.List <- epihet::makeGR(files = files, ids = ids,
cores = 1, sve = FALSE)
```

`moduleAnno` *module annotation*

Description

annotate modules using differentially expressed genes

Usage

```
moduleAnno(DEG, background, module.gene, cutoff = 0.05,
  adjust.method = "fdr", prefix = NA, pdf.height = 10,
  pdf.width = 10, sve = FALSE)
```

Arguments

`DEG`	a character vector containing up/down regulated genes
`background`	a charactor vector containing all genes as background in hypergeometric test
`module.gene`	a data frame containing genes with genome region containing DEH loci from one module, generated from epiNetwork() function. The first column is gene entrez ID, the second column is module lable, the third column is module color
`cutoff`	Cutoff value of qvalue for gene enrichment (default: 0.05)
`adjust.method`	one of 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr', 'none'(default:fdr)
`prefix`	a prefix for PDF file name
`pdf.height`	An integer representing the height (in inches) of the outputted boxplot pdf file (default: 10)
`pdf.width`	An integer representing the width (in inches) of the outputted boxplot pdf file (default: 10)
`sve`	A boolean to save the plot (default: FALSE)

Value

a data frame showing modules that were enriched by DEGs and module size, p value and q value

Examples

```
data(DEG,package = "epihet")
data(background,package = "epihet")
module<-data.frame(gene=c("NM_000014","NM_000015","NM_000017","NM_000019",
"NM_052960","NR_138250","NM_000037","NM_000038","NM_000039","NM_000044",
"NM_000046","NM_015074","NM_183416","NM_004421","NM_001330311",
"NM_001145210","NM_000097","NM_000103","NM_000104",
"NM_000079","NM_000083","NM_000086","NM_000087","NM_000092","NM_000094",
"NM_000095","NM_006474"),label=rep(c(1,2),c(12,15)),
color=rep(c("purple","brown"),c(12,15)),
stringsAsFactors = FALSE)
module.annotation<-epihet::moduleAnno(DEG$refseq,background$gene,
                                      module.gene=module,
                                      cutoff=0.05,adjust.method = "fdr",
                                      prefix='epipoly',pdf.height = 10,
                                      pdf.width = 10, sve = TRUE)
```

moduleSim *module comparison between two subtypes*

Description

Compare any two modules from two subytpes based on genes shared by the modules

Usage

```
moduleSim(module.subtype1, module.subtype2, pdf.height = 10,
  pdf.width = 10, sve = FALSE)
```

Arguments

`module.subtype1`
a data frame generated from the epiNetwork() function the module information of subtype1,the first column is module nodes,the second column is module label, the third column is module color

`module.subtype2`
a data frame generated from the epiNetwork() function. The module information of subtype1, the first column is module nodes, the second column is module label, the third column is module color

`pdf.height` An integer representing the height (in inches) of the outputted boxplot pdf file (default: 10)

`pdf.width` An integer representing the width (in inches) of the outputted boxplot pdf file (default: 10)

`sve` A boolean to save the plot (default: FALSE)

Value

a matrix containing Jaccard scores

Examples

```
data(modulesil,package = "epihet")
data(moduledm,package = "epihet")
sim.score<-epihet::moduleSim(module.subtype1=modulesil,
                             module.subtype2=moduledm,
                             pdf.height = 10,pdf.width = 10,
                             sve = TRUE)
```

---

moduleVisual *Modules visualization and network topology*

Description

Visualize the modules identified by epiNetwork() function, and calculate network topology

Usage

```
moduleVisual(TOM, value.matrix, moduleColors, mymodule, cutoff = 0.02,
  prefix = NULL, sve = FALSE)
```

Arguments

TOM	the topological overlap matrix in WGCNA generated from the epiNetwork() function
value.matrix	A data frame generated from the epiNetwork() function. the row name is patients in one subtype. the column name is the DEH loci the value in the matrix is epigenetic heterogeneity on one DEH loci for one patient
moduleColors	the module assignment generated from the epiNetwork() function
mymodule	a character vector containing the module colors you want to visulaize
cutoff	adjacency threshold for including edges in the output (default:0.02)
prefix	a character for output filename
sve	A boolean to save the plot (default: FALSE)

Value

a list containing all module edge and node information for mymodule

Examples

```
correlation.m<-matrix(0,12,12)
correlation.m[1,c(2:10)]<-c(0.006,0.054,0.079,0.078, 0.011,0.033,0.014,
0.023,0.034)
correlation.m[2,c(3:10)]<-c(0.026,0.014,0.045,0.037, 0.026,0.011,0.034,
0.012)
correlation.m[3,c(4:10)]<-c(0.016,0.024,0.039,0.045, 0.009,0.003,0.028)
correlation.m[4,c(5:10)]<-c(0.039,0.002,0.053,0.066, 0.012,0.039)
correlation.m[5,c(6:10)]<-c(0.019,0.016,0.047,0.046, 0.013)
correlation.m[6,c(7:10)]<-c(0.017,0.057,0.029,0.056)
correlation.m[7,c(8:10)]<-c(0.071,0.018,0.001)
correlation.m[8,c(9:10)]<-c(0.046,0.014)
correlation.m[9,10]<-0.054
correlation.m[lower.tri(correlation.m)] <-
t(correlation.m)[lower.tri(correlation.m)]

matrix.v<-matrix(0.5,5,12)
matrix.v<-as.data.frame(matrix.v)
colnames(matrix.v)<-c("NM_052960","NR_138250","NM_015074","NM_183416",
"NM_017891","NM_001330306","NM_014917","NM_001312688","NM_001330665",
"NM_017766","NM_001079843","NM_001040709")
modulecolor<-c(rep(c("yellow","cyan"),c(10,2)))
module.topology<-epihet::moduleVisual(correlation.m,
                                      value.matrix=matrix.v,
                                      moduleColors=modulecolor,
                                      mymodule="yellow",cutoff=0.02,
                                      prefix='CEBPA_sil_epipoly',sve = TRUE)
```

readGR *Make GenomicRanges Object*

Description

Creates a GenomicRanges file for a singular methclone ouput file

Usage

```
readGR(files, ids, n)
```

Arguments

files	A vector of files containing methcolone output
ids	A vector of sample ids for the files
n	The index of the file vector to be read

Value

A GenomicRanges object containing pdr, epipolymorphism, and Shannon entropy values for the nth file

Examples

```
files <- c(system.file("extdata","D-2238.chr22.region.methClone_out.gz",package = "epihet"),
system.file("extdata","D-2668.chr22.region.methClone_out.gz",package = "epihet"),
system.file("extdata","N-1.chr22.region.methClone_out.gz",package = "epihet"),
system.file("extdata","N-2.chr22.region.methClone_out.gz",package = "epihet"))
ids <- epihet::splitn(basename(files),"[.]",1)
GR.Object <- epihet::readGR(files = files, ids = ids, n = 3)
```

shannon *Shannon Entropy*

Description

Calculates the Shannon entropy value

Usage

```
shannon(p)
```

Arguments

p	A vector of epiallele probabilities

Value

The Shannon entropy value

Examples

```
a<-c(rep(0,13),60.86960,0.00000,39.1304)
shannon(a)
```

splitn *The subtring extraction of a character vector*

Description

Extract the subtrings of a character vector according to the matches to substring split within them.

Usage

```
splitn(strings, field, n)
```

Arguments

strings	A GenomicRanges object to be compared
field	A GenomicRanges object to be compared
n	The value of gr1 to be compared

Value

A data frame containing a summary of the GenomicRanges object

Examples

```
x<-'chr1:10000-10005'
splitn(x,':',1)
```

summarize *Summarize Data*

Description

Summarizes pdr, epipolymorphism, and shannon values over the annotation regions

Usage

```
summarize(gr1, gr2, value1, value2, cutoff1 = 10, cutoff2 = 60)
```

Arguments

`gr1`	A GenomicRanges object to be compared
`gr2`	A GenomicRanges object to be compared
`value1`	The value of gr1 to be compared
`value2`	The value of gr2 be compared
`cutoff1`	The first cutoff value for the number of reads (default:10)
`cutoff2`	The second cutoff value for the number of reads (default:60)

Value

A data frame containing a summary of the GenomicRanges object

Examples

```
p1.GR<-GRanges(seqnames = Rle(c("chr22"), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c("-", "+", "+", "+", "-"))),
values.loci = c("327:350:361:364","821:837:844:849",
"838:845:850:858","755:761:771:773","761:771:773:781"),
values.read1 = c(92,72,68,176,176),values.meth1=c(84,93,94,96,95),
values.shannon=c(0.4,0.5,0.5,0.2,0.5),values.pdr=c(0.6,0.25,0.23,0.15,0.17),
values.epipoly=c(0.48,0.42,0.38,0.27,0.3))

p2.GR<-GRanges(seqnames = Rle(c("chr22"), c(5)),
ranges = IRanges(c(327,821,838,755,761), end = c(364,849,858,773,781)),
strand = Rle(strand(c("-", "+", "+", "+", "-"))),
values.loci = c("327:350:361:364","821:837:844:849",
"838:845:850:858","755:761:771:773","761:771:773:781"),
values.read1 = c(107,102,102,76,76),values.meth1=c(88,66,69,71,94),
values.shannon=c(0.12,0.25,0.54,0.23,0.25),
values.pdr=c(0.38,1,0.97,1,0.13),
values.epipoly=c(0.57,0.42,0.28,0.18,0.23))

GR.List<-list(p1=p1.GR,p2=p2.GR)
summary <- summarize(gr1 = GR.List[[1]], gr2 = GR.List[[2]],
value1 = 'pdr', value2 = 'epipoly',
cutoff1 = 10, cutoff2 = 60)
```

userobj *GenomicRanges object generation*

Description

generate GenomicRanges object for DEH loci

Usage

```
userobj(data)
```

Arguments

data a data frame containing the chromosome number, loci and strand information of DEH loci generated from diffHet() function.

Value

A GenomicRanges object

Examples

```
data<-data.frame(chromosome=c('chr1','chr1','chr1'),
loci=c('6480531:6480554:6480561:6480565','6647655:6647696:6647701:6647705',
'7130155:7130172:7130179:7130188'),
strand=c('+','-','+'),stringsAsFactors = FALSE)
userobj(data)
```

Index