Package 'epimutacions'

October 21, 2025

Title Robust outlier identification for DNA methylation data

Version 1.13.0

Description The package includes some statistical outlier detection methods for epimutations detection in DNA methylation data.
The methods included in the package are MANOVA, Multivariate linear models, isolation forest, robust mahalanobis distance, quantile and beta.
The methods compare a case sample with a suspected disease against a reference panel (composed of healthy individuals) to identify epimutations in the given case sample.
It also contains functions to annotate and visualize the identified epimutations.

biocViews DNAMethylation, BiologicalQuestion, Preprocessing, StatisticalMethod, Normalization

License MIT + file LICENSE

Depends R (>= 4.3.0), epimutacionsData

Imports minfi, bumphunter, isotree, robustbase, ggplot2, GenomicRanges, GenomicFeatures, IRanges, SummarizedExperiment, stats, matrixStats, BiocGenerics, S4Vectors, utils, biomaRt, BiocParallel, GenomeInfoDb, Homo.sapiens, purrr, tibble, Gviz, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg18.knownGene, TxDb.Hsapiens.UCSC.hg38.knownGene, rtracklayer, AnnotationDbi, AnnotationHub, ExperimentHub, reshape2, grid, ensembldb, gridExtra, IlluminaHumanMethylation450kmanifest, IlluminaHumanMethylationEPICmanifest, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylationEPICanno.ilm10b2.hg19, ggrepel

Suggests testthat, knitr, rmarkdown, BiocStyle, a4Base, kableExtra, methods, grDevices

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/isglobal-brge/epimutacions

BugReports https://github.com/isglobal-brge/epimutacions/issues

RoxygenNote 7.2.3

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

git_branch devel

git_last_commit fa459c5

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-10-21

Author Dolors Pelegri-Siso [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5993-3003>),
Juan R. Gonzalez [aut] (ORCID: <https://orcid.org/0000-0003-3267-2146>),
Carlos Ruiz-Arenas [aut] (ORCID:
<https://orcid.org/0000-0002-6014-3498>),
Carles Hernandez-Ferrer [aut] (ORCID:
<https://orcid.org/0000-0002-8029-7160>),
Leire Abarrategui [aut] (ORCID:
<https://orcid.org/0000-0002-1175-038X>)

Maintainer Dolors Pelegri-Siso <dolors.pelegri@isglobal.org>

Contents

`add_ensemble_regulatory` *Add ENSEMBL regulatory regions to epimutations*

Description

Add ENSEMBL regulatory regions to epimutations

Usage

```
add_ensemble_regulatory(epimutations, build = "37")
```

Arguments

`epimutations`	a data frame object containing the result from `epimutations` or `epimutations_one_leave_out` functions.
`build`	the build used to define epimutations coordinates. By default, it is `'37'`, corresponding to Illumina annotation.

Value

The function returns a data frame object containing the results of `epimutations` or `epimutations_one_leave_out` with some additional variables describing regulatory elements from ENSEMBL.

Note that a single epimutation might overlap with more than one regulatory region. In that case, the different regulatory regions are separated by `///`.

- ensembl_reg_idRegion identifier from ENSEMBL
- ensembl_reg_coordinatesCoordinates for the ENSEMBL regulatory regions
- ensembl_reg_typeType of regulatory region
- ensembl_reg_tissuesActivity of the regulatory region per tissue. The different activation states are separated by `/`

`annotate_cpg` *Annotate the DMR resulting from epimutacions package*

Description

This function annotates a differentially methylated region

Usage

```
annotate_cpg(
  data,
  db,
  split = ",",
  epi_col = "cpg_ids",
  gene_col = "GencodeBasicV12_NAME",
  feat_col = "Regulatory_Feature_Group",
  relat_col = "Relation_to_Island",
  build = "37",
  omim = TRUE
)
```

Arguments

data	DataFrame-like object.
db	a character string specifying the Database to use for annotation. E.g: 'IlluminaHumanMethylationE
split	a character string containing the separator for CpG ids. Default ','.
epi_col	CpG ids, should be row names in the data base.
gene_col	column name from where to extract gene names. Default: 'GencodeBasicV12_NAME'.
feat_col	column name from where to extract CpG feature groups. Default: 'Regulatory_Feature_Group'.
relat_col	column name from where to extract relation to island info. Default: 'Relation_to_Island'.
build	The build for bioMart. Default '37'.
omim	a boolean, if TRUE will annotate OMIMs as well. Takes a bit longer. Default TRUE.

Value

The function returns a DataFrame-like object annotated.

---

annotate_epimutations *Annotate the results of epimutations or epimutations_one_leave_out functions*

Description

Information about close genes and regulatory elements for epimutations.

Usage

```
annotate_epimutations(
  epi_results,
  db = "IlluminaHumanMethylationEPICanno.ilm10b2.hg19",
  build = "37",
  ...
)
```

Arguments

epi_results	a data frame object containing the output from epimutations or epimutations_one_leave_out functions.
db	a character string containing the Illumina annotation package used to annotate the CpGs.
build	a character string containing the genomic build where the epimutations are mapped. The default is GRCh37 (build = "37"). To use GRCh38 set built to NULL.
...	Further arguments passed to annotate_cpg.

Value

The function returns the input object epi_results with additional columns containing the information about the genes or overlapping regulatory features.

See annotate_cpg and add_ensemble_regulatory for an in-depth description of these variables.

Examples

```
data(res.epi.manova)
#Annotate the epimutations

#anno_results <- annotate_epimutations(res.epi.manova)
```

AS204 *Algorithm AS 204*

Description

Distribution of a positive linear combination of χ^2 random variables.

Usage

```
AS204(
  c,
  lambda,
  mult = rep(1, length(lambda)),
  delta = rep(0, length(lambda)),
  maxit = 1e+05,
  eps = 1e-14,
  mode = 1
)
```

Arguments

c	value point at which distribution is to be evaluated.
lambda	the weights λ_j.
mult	the multiplicities m_j.
delta	the non-centrality parameters δ_j^2.
maxit	the maximum number of terms K (see Details).
eps	the desired level of accuracy.
mode	if "mode" > 0 then $\beta = mode\lambda_{min}$, otherwise $\beta = 2/(1/\lambda_{min} + 1/\lambda_{max})$.

Details

Algorithm AS 204 evaluates the expression

$$P[X < c] = P[\sum_{j=1}^{n} \lambda_j \chi^2(m_j, \delta_j^2) < c]$$

where λ_j and c are positive constants and $\chi^2(m_j, \delta_j^2)$ represents an independent χ^2 random variable with m_j degrees of freedom and non-centrality parameter δ_j^2. This can be approximated by the truncated series

$$\sum_{k=0}^{K-1} a_k P[\chi^2(m + 2k) < c/\beta]$$

where $m = \sum_{j=1}^{n} m_j$ and β is an arbitrary constant (as given by argument "mode").

The C++ implementation of algorithm AS 204 used here is identical to the one employed by the farebrother method in the CompQuadForm package, with minor modifications.

Value

The function returns the probability $P[X > c] = 1 - P[X < c]$ if the AS 204 fault indicator is 0 (see Note below), and NULL if the fault indicator is 4, 5 or 9, as the corresponding faults can be corrected by increasing "eps". Other faults raise an error.

Note

The algorithm AS 204 defines the following fault indicators: **-j)** one or more of the constraints $\lambda_j > 0$, $m_j > 0$ and $\delta_j^2 \geq 0$ is not satisfied. **1)** non-fatal underflow of a_0. **2)** one or more of the constraints $n > 0$, $c > 0$, $maxit > 0$ and $eps > 0$ is not satisfied. **3)** the current estimate of the probability is < -1. **4)** the required accuracy could not be obtained in $maxit$ iterations. **5)** the value returned by the procedure does not satisfy $0 \leq P[X < c] \leq 1$. **6)** the density of the linear form is negative. **9)** faults 4 and 5. **10)** faults 4 and 6. **0)** otherwise.

Author(s)

Diego Garrido-Martín

References

P. Duchesne, P. Lafaye de Micheaux, Computing the distribution of quadratic forms: Further comparisons between the Liu-Tang-Zhang approximation and exact methods, Computational Statistics and Data Analysis, Vol. 54, (2010), 858-862

Farebrother R.W., Algorithm AS 204: The distribution of a Positive Linear Combination of chi-squared random variables, Journal of the Royal Statistical Society, Series C (applied Statistics), Vol. 33, No. 3 (1984), 332-339

See Also

farebrother

`betas_from_bump` *Obtains bumps beta values*

Description

The function obtains beta values corresponding to the CpGs into DMRs.

Usage

```
betas_from_bump(bump, fd, betas)
```

Arguments

`bump`	the result from bumphunter.
`fd`	a data frame containing the genomic ranges for each CpGs.
`betas`	a matrix containing the beta values for all CpGs in each sample.

Value

The function returns a data frame containing the beta values for each sample and CpG into DMR.

`betas_sd_mean` *Computes beta values, standard deviation and mean to plot the epimutation*

Description

Computes the beta values, population mean and 1, 1.5, and 2 standard deviations from the mean of the distribution necessary to plot the epimutations.

Usage

```
betas_sd_mean(gr)
```

Arguments

`gr`	a GRanges object obtained from create_GRanges_class function.

Value

The function returns a list containing the melted beta values, the population mean and 1, 1.5, and 2 standard deviations from the mean of the distribution.

cols_names *Sets common column names in a data frame*

Description

Sets common column names in a given data frame containing the CpGs genomic ranges or a DMR (result of epimutations or epimutations_one_leave_out function).

Usage

```
cols_names(x, cpg_ids_col = FALSE)
```

Arguments

x	a data frame containing the genomic ranges or a DMR (a row of the results of epimutations or epimutations_one_leave_out function).
cpg_ids_col	a boolean, if TRUE the input data frame contains the CpGs names column.

Value

The function returns a data frame containing the column names to carry out the analysis without any error.

create_GRanges_class *Generates a GRanges object*

Description

This function makes a GRanges object from a GenomicRatioSet.

Usage

```
create_GRanges_class(methy, cpg_ids)
```

Arguments

methy	a GenomicRatioSet object containing the control and case samples used in epimutations or epimutations_one_leave_out function.
cpg_ids	a character string specifying the name of the CpGs in the DMR of interest.

Value

The function returns a GRanges object containing the beta values and the genomic ranges of the CpGs of interest.

epimutations *Epimutations analysis based on outlier detection methods*

Description

The function identifies differentially methylated regions in a case sample by comparing it against a control panel.

Usage

```
epimutations(
  case_samples,
  control_panel,
  method = "manova",
  chr = NULL,
  start = NULL,
  end = NULL,
  epi_params = epi_parameters(),
  maxGap = 1000,
  bump_cutoff = 0.1,
  min_cpg = 3,
  verbose = TRUE
)
```

Arguments

case_samples	a GenomicRatioSet object containing the case samples. See the constructor function GenomicRatioSet, makeGenomicRatioSetFromMatrix.
control_panel	a GenomicRatioSet object containing the control panel (control panel).
method	a character string naming the outlier detection method to be used. This can be set as: "manova", "mlm", "iForest", "mahdist", "quantile" and "beta". The default is "manova". For more information see **Details**.
chr	a character string containing the sequence names to be analysed. The default value is NULL.
start	an integer specifying the start position. The default value is NULL.
end	an integer specifying the end position. The default value is NULL.
epi_params	the parameters for each method. See the function epi_parameters.
maxGap	the maximum location gap used in bumphunter method.
bump_cutoff	a numeric value of the estimate of the genomic profile above the cutoff or below the negative of the cutoff will be used as candidate regions.
min_cpg	an integer specifying the minimum CpGs number in a DMR.
verbose	logical. If TRUE additional details about the procedure will provide to the user. The default is TRUE.

Details

The function compares a case sample against a control panel to identify epimutations in the given sample. First, the DMRs are identified using the bumphunter approach. After that, CpGs in those DMRs are tested in order to detect regions with CpGs being outliers. For that, different outlier detection methods can be selected:

- Multivariate Analysis of Variance (`"manova"`). manova
- Multivariate Linear Model (`"mlm"`)
- Isolation Forest (`"iForest"`) isolation.forest
- Robust Mahalanobis Distance (`"mahdist"`) covMcd
- Quantile distribution (`"quantile"`)
- Beta (`"beta"`)

We defined candidate epimutation regions (found in candRegsGR) based on the 450K array design. As CpGs are not equally distributed along the genome, only CpGs closer to other CpGs can form an epimutation. More information can be found in candRegsGR documentation.

Value

The function returns an object of class tibble containing the outliers regions. The results are composed by the following columns:

- `epi_id`: systematic name for each epimutation identified. It provides the name of the used anomaly detection method.
- `sample`: the name of the sample containing the epimutation.
- `chromosome`, `start` and `end`: indicate the location of the epimutation.
- `sz`: the window's size of the event.
- `cpg_n`: the number of CpGs in the epimutation.
- `cpg_ids`: the names of CpGs in the epimutation.
- `outlier_score`:
  - For method `manova` it provides the approximation to F-test and the Pillai score, separated by `/`.
  - For method `mlm` it provides the approximation to F-test and the R2 of the model, separated by `/`.
  - For method `iForest` it provides the magnitude of the outlier score.
  - For method `beta` it provides the mean outlier p-value.
  - For methods `quantile` and `mahdist` it is filled with NA.
- `outlier_direction`: indicates the direction of the outlier with `"hypomethylation"` and `"hypermethylation"`
  - For `manova`, `mlm`, `iForest`, and `mahdist` it is computed from the values obtained from bumphunter.
  - For `quantile` it is computed from the location of the sample in the reference distribution (left vs. right outlier).
  - For method `beta` it return a NA.
- `pvalue`:
  - For methods `manova`, `mlm`, and `iForest` it provides the p-value obtained from the model.
  - For method `quantile`, `mahdist` and `beta` is filled with NA.

- adj_pvalue: for methods with p-value (manova and mlm adjusted p-value with Benjamini-Hochberg based on the total number of regions detected by Bumphunter.
- epi_region_id: Name of the epimutation region as defined in candRegsGR.
- CRE: cREs (cis-Regulatory Elements) as defined by ENCODE overlapping the epimutation region. Different cREs are separated by ;.
- CRE_type: Type of cREs (cis-Regulatory Elements) as defined by ENCODE. Different type are separeted by, and different cREs are separated by ;.

Examples

```
data(GRset)

#Find epimutations in GSM2562701 sample of GRset dataset

case_samples <- GRset[,11]
control_panel <- GRset[,1:10]
epimutations(case_samples, control_panel, method = "manova")
```

---

epimutations_one_leave_out

Epimutations analysis based on outlier detection methods

Description

This function is similar to epimutations with the particularity that when is more than one case sample, the remaining case samples are included as controls.

Usage

```
epimutations_one_leave_out(
  methy,
  method = "manova",
  epi_params = epi_parameters(),
  BPPARAM = BiocParallel::SerialParam(),
  verbose = TRUE,
  ...
)
```

Arguments

methy	a GenomicRatioSet object containing the samples for the analysis. See the constructor function GenomicRatioSet, makeGenomicRatioSetFromMatrix.
method	a character string naming the outlier detection method to be used. This can be set as: "manova", "mlm", "iForest", "mahdist", "barbosa" and beta. The default is "manova". For more information see **Details**.
epi_params	the parameters for each method. See the function epi_parameters.
BPPARAM	("BiocParallelParam") BiocParallelParam object to configure parallelization execution. By default, execution is non-parallel.
verbose	logical. If TRUE additional details about the procedure will provide to the user. The default is TRUE.
...	Further parameters passed to epimutations

Details

The function compares a case sample against a control panel to identify epimutations in the given sample. First, the DMRs are identified using the bumphunter approach. After that, CpGs in those DMRs are tested in order to detect regions with CpGs being outliers. For that, different anomaly detection methods can be selected:

- Multivariate Analysis of Variance (`"manova"`). manova
- Multivariate Linear Model (`"mlm"`)
- Isolation Forest (`"iForest"`) isolation.forest
- Robust Mahalanobis Distance (`"mahdist"`) covMcd
- Barbosa (`"barbosa"`)

Value

The function returns an object of class tibble containing the outliers regions. The results are composed by the following columns:

- `epi_id`: the name of the anomaly detection method that has been used to detect the epimutation
- `sample`: the name of the sample where the epimutation was found.
- `chromosome`, `start` and `end`: indicate the location of the epimutation.
- `sz`: the number of base pairs in the region.
- `cpg_n`: number of CpGs in the region.
- `cpg_ids`: differentially methylated CpGs names.
- `outlier_score`:
  - For method `manova` it provides the approximation to F-test and the Pillai score, separated by `/`.
  - For method `mlm` it provides the approximation to F-test and the R2 of the model, separated by `/`.
  - For method `iForest` it provides the magnitude of the outlier score.
  - For methods `barbosa` and `mahdist` is filled with NA.
- `outlier_significance`:
  - For methods `manova`, `mlm`, and `iForest` it provides the p-value obtained from the model.
  - For method `barbosa` and `mahdist` is filled with NA.
- `outlier_direction`: indicates the direction of the outlier with `"hypomethylation"` and `"hypermethylation"`
  - For `manova`, `mlm`, `iForest`, and `mahdist` it is computed from the values obtained from bumphunter.
  - For `barbosa` it is computed from the location of the sample in the reference distribution (left vs. right outlier).

Examples

```
data(GRset)
manova_result <- epimutations_one_leave_out(GRset,
                                            method = "manova")
```

epi_beta *Identifies epimutations based on a beta distribution.*

Description

epi_beta method models the DNA methylation data using a beta distribution. First, the beta distribution parameters of the reference population are precomputed and passed to the method. Then, we compute the probability of observing the methylation values of the case from the reference beta distribution. CpGs with p-values smaller than a threshold pvalue_threshold and with a methylation difference with the mean reference methylation higher than diff_threshold are defined as outlier CpGs. Finally, epimutations are defined as a group of contiguous outlier CpGs.

Usage

```
epi_beta(
  beta_params,
  beta_mean,
  betas_case,
  case,
  controls,
  betas,
  annot,
  pvalue_threshold,
  diff_threshold,
  min_cpgs = 3,
  maxGap
)
```

Arguments

beta_params	matrix with the parameters of the reference beta distributions for each CpG in the dataset.
beta_mean	beta values mean.
betas_case	matrix with the methylation values for a case.
case	case sample name.
controls	control samples names.
betas	a matrix containing the beta values for all samples.
annot	annotation of the CpGs.
pvalue_threshold	minimum p-value to consider a CpG an outlier.
diff_threshold	minimum methylation difference between the CpG and the mean methylation to consider a position an outlier.
min_cpgs	minimum number of CpGs to consider an epimutation.
maxGap	maximum distance between two contiguous CpGs to combine them into an epimutation.

Value

The function returns a data frame with the candidate regions to be epimutations.

epi_iForest *Identifies epimutations using Isolation Forest*

Description

This function identifies regions with CpGs being outliers using isolation.forest approach.

Usage

```
epi_iForest(mixture, case_id, ntrees)
```

Arguments

mixture	beta values matrix. Samples in columns and CpGs in rows.
case_id	a character string specifying the name of the case sample.
ntrees	number of binary trees to build for the model. Default is 100.

Value

The function returns the outlier score for the given case sample.

epi_mahdist *Identifies epimutations using Robust Mahalanobis distance*

Description

This function identifies regions with CpGs being outliers using the Minimum Covariance Determinant (MCD) estimator (covMcd) to compute the Mahalanobis distance.

Usage

```
epi_mahdist(mixture, nsamp = c("best", "exact", "deterministic"))
```

Arguments

mixture	beta values matrix. Samples in columns and CpGs in rows.
nsamp	the number of subsets used for initial estimates in the MCD. It can be set as: "best", "exact", or "deterministic".

Details

The implementation of the method here is based on the discussion in this thread of Cross Validated

Value

The function returns the computed Robust Mahalanobis distance.

epi_manova *Identifies epimutations using MANOVA*

Description

This function identifies regions with CpGs being outliers using manova approach.

Usage

```
epi_manova(mixture, model, case_id)
```

Arguments

mixture	beta values matrix. Samples in columns and CpGs in rows.
model	design (or model) matrix.
case_id	a character string specifying the name of the case sample.

Value

The function returns the F statistic, Pillai and P value.

epi_mlm *Detects epimutations using Multivariate Linear Model (MLM)*

Description

Identifies CpGs with outlier methylation values using methylated Multivariate Linear Model

Usage

```
epi_mlm(mixture, model)
```

Arguments

mixture	beta values matrix. Samples in columns and CpGs in rows.
model	design (or model) matrix.

Value

The function returns the F statistic, R2 test statistic and Pillai.

epi_parameters *Settings for parameters of* epimutations *and* epimutations_one_leave_out *functions*

Description

Allow the user to set the values of the parameters to compute the functions epimutations and epimutations_one_leave_out.

Usage

```
epi_parameters(
  manova = list(pvalue_cutoff = 0.05),
  mlm = list(pvalue_cutoff = 0.05),
  iForest = list(outlier_score_cutoff = 0.7, ntrees = 100),
  mahdist = list(nsamp = "deterministic"),
  quantile = list(window_sz = 1000, offset_abs = 0.15, qsup = 0.995, qinf = 0.005),
  beta = list(pvalue_cutoff = 1e-06, diff_threshold = 0.1)
)
```

Arguments

manova, mlm, iForest, mahdist, quantile, beta
: method selected in the function epimutations.

pvalue_cutoff
: the threshold p value to select which CpG regions are outliers in manova, mlm and beta methods.

outlier_score_cutoff
: The outlier score threshold to identify outliers CpGs in isolation forest (iForest) method. Default is 0.5.

ntrees
: number of binary trees to build for the model build by isolation forest (iForest) method. Default is 100.

nsamp
: the number of subsets used for initial estimates in the Minimum Covariance Determinant which is used to compute the Robust Mahalanobis distance (mahdist). It can be set as: "best", "exact", or "deterministic". For nsamp = "best" exhaustive enumeration is done, as long as the number of trials does not exceed 100'000. For nsamp = "exact" exhaustive enumeration will be attempted however many samples are needed. In this case, a warning message may be displayed saying that the computation can take a very long time. For nsamp = "deterministic". For more information see covMcd. Default is "deterministic".

window_sz
: the maximum distance between CpGs to be considered in the same DMR. This parameter is used in quantile (default: 1000).

qsup, qinf, offset_abs
: The upper and lower quantiles (threshold) to consider a CpG an outlier when using quantile method, as well as the offset to consider (defaults: 0.005, 0.995, 0.15).

diff_threshold
: Minimum methylation difference between the CpG and the mean methylation to consider a position an outlier.

Details

Invoking `epi_parameters()` with no arguments returns return a list with the default values.

Value

the function returns a list of all set parameters for each method used in epimutations and epimutations_one_leave_out functions.

Examples

```
#Default set of parameters
epi_parameters()
#change p value for manova method
epi_parameters(manova = list("pvalue_cutoff" = 0.01))
```

`epi_preprocess` *Preprocess methylation array*

Description

The `epi_preprocess` function reads Illumina methylation sample sheet for case samples and it merges them with RGChannelSet reference panel. The final dataset is normalized using minfi package preprocess methods.

Usage

```
epi_preprocess(
  cases_dir,
  reference_panel,
  pattern = "csv$",
  normalize = "raw",
  norm_param = norm_parameters(),
  verbose = FALSE
)
```

Arguments

`cases_dir`	the base directory from which the search is started.
`reference_panel`	an RGChannelSet object containing the reference panel (controls) samples.
`pattern`	What pattern is used to identify a sample sheet file.
`normalize`	a character string specifying the selected preprocess method. For more information see **Details** or minfi package user's Guide. It can be set as: `"raw"`, `"illumina"`, `"swan"`, `"quantile"`, `"noob"` or `"funnorm"`.)
`norm_param`	the parameters for each preprocessing method. See the function norm_parameters.
`verbose`	logical. If TRUE additional details about the procedure will provide to the user. The default is FALSE.

Details

The epi_preprocess function reads Illumina methylation sample sheet for case samples and it merges them with RGChannelSet reference panel. The final dataset is normalized using different minfi package preprocess methods:

- "raw": preprocessRaw
- "illumina": preprocessIllumina
- "swan": preprocessSWAN
- "quantile": preprocessQuantile
- "noob": preprocessNoob
- "funnorm": preprocessFunnorm

Value

epi_preprocess function returns a GenomicRatioSet object containing case and control (reference panel) samples.

Examples

```
# The reference panel for this example is available in
#epimutacionsData (ExperimentHub) package

   library(ExperimentHub)
   eh <- ExperimentHub()
   query(eh, c("epimutacionsData"))
   reference_panel <- eh[["EH6691"]]
   cases_dir <- system.file("extdata", package = "epimutacionsData")
   #Preprocessing

   epi_preprocess( cases_dir,
                   reference_panel,
                   pattern = "SampleSheet.csv")
```

---

epi_quantile *Identifies epimutations using quantile distribution*

Description

Identifies CpGs with outlier methylation values using a sliding window approach to compare individual methylation profiles of a single case sample against all other samples from reference panel (controls)

Usage

```
epi_quantile(
  case,
  fd,
  bctr_pmin,
  bctr_pmax,
  controls,
  betas,
  window_sz = 1000,
  N = 3,
  offset_abs = 0.15
)
```

Arguments

case	beta values for a single case (data.frame). The samples as single column and CpGs in rows (named).
fd	feature description as data.frame having at least chromosome and position as columns and and CpGs in rows (named).
bctr_pmin	Beta value observed at 0.01 quantile in controls. A beta values has to be lower or equal to this value to be considered an epimutation.
bctr_pmax	Beta value observed at 0.99 quantile in controls. A beta values has to be higher or equal to this value to be considered an epimutation.
controls	control samples names.
betas	a matrix containing the beta values for all samples.
window_sz	Maximum distance between a pair of CpGs to defined an region of CpGs as epimutation (default: 1000).
N	Minimum number of CpGs, separated in a maximum of window_sz bass, to defined an epimutation (default: 3).
offset_abs	Extra enforcement defining an epimutation based on beta values at 0.005 and 0.995 quantiles (default: 0.15).

Value

The function returns a data frame with the regions candidates to be epimutations.

---

getBetaParams *Model methylation as a beta distribution*

Description

Model methylation as a beta distribution

Usage

```
getBetaParams(x)
```

Arguments

x	Matrix of methylation expressed as a beta. CpGs are in columns and samples in rows.

Value

Beta distribution.

---

get_candRegsGR *Candidate regions to be epimutations*

Description

Load candidate regions to be epimutations from epimutacionsData package in ExperimentHub.

Usage

```
get_candRegsGR()
```

Value

The function returns a GRanges object containing the candidate regions.

---

get_ENSEMBL_data *Get ENSEMBL regulatory features overlapping a genomic region*

Description

This function queries for ENSEMBL regulatory features and collapse them to return a single record.

Usage

```
get_ENSEMBL_data(chromosome, start, end, mart)
```

Arguments

chromosome	Chromosome of the region
start	Start of the region
end	End of the region
mart	Mart object to perform the ENSEMBL query

Value

data.frame of one row with the ENSEMBL regulatory regions overlapping the genomic coordinate.

GRset *GRset*

Description

A small GenomicRatioSet object to use in the functions examples containing 10 control samples and a case sample.

Usage

```
data(GRset)
```

Format

A GenomicRatioSet object with 4243 CpGs and 11 variables

Value

A GenomicRatioSet object with 4243 CpGs and 11 variables

Examples

```
data(GRset)
```

merge_records *Merge records for the same ENSEMBL regulatory element*

Description

This function collapses the activity status of a given an ENSEMBL regulatory element in different tissues. Notice that tissues identified as inactive will not be reported.

Usage

```
merge_records(tab)
```

Arguments

tab	Results from `biomaRt::getBM` for the same regulatory element

Value

`data.frame` of one row after collapsing the

`mlm` *Non-parametric, Asymptotic P-values for Multivariate Linear Models*

Description

Fits a multivariate linear model and computes test statistics and asymptotic P-values for predictors in a non-parametric manner.

Usage

```
mlm(
  formula,
  data,
  transform = "none",
  contrasts = NULL,
  subset = NULL,
  fit = FALSE
)
```

Arguments

`formula`	object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `mlm` is called.
`transform`	transformation of the response variables: "`none`", "`sqrt`" or "`log`". Default is "`none`".
`contrasts`	an optional list. See `contrasts.arg` in `model.matrix.default`. Default is "`contr.sum`" for ordered factors and "`contr.poly`" for unordered factors. Note that this is different from the default setting in `options("contrasts")`.
`subset`	subset of predictors for which summary statistics will be reported. Note that this is different from the "`subset`" argument in `lm`.
`fit`	logical. If `TRUE` the multivariate fit on transformed and centered responses is returned.

Details

A `Y` matrix is obtained after transforming (optionally) and centering the original response variables. Then, the multivariate fit obtained by `lm` can be used to compute sums of squares, pseudo-F statistics and asymptotic P-values for the terms specified by the `formula` in a non-parametric manner.

Value

`mlm` returns an object of `class` "`MLM`", a list containing:

`call`	the matched call.
`aov.tab`	ANOVA table with Df, Sum Sq, Mean Sq, F values, partial R-squared and P-values.

precision	the precision in P-value computation.
transform	the transformation applied to the response variables.
na.omit	incomplete cases removed (see na.omit).
fit	if fit = TRUE the multivariate fit done on the transformed and centered response variables is also returned.

Author(s)

Diego Garrido-Martín

See Also

lm, Anova

---

mlmtst *Sums of Squares and Pseudo-F Statistics from a Multivariate Fit*

Description

Computes the sum of squares, degrees of freedom, pseudo-F statistics and partial R-squared for each predictor from a multivariate fit. It also returns the eigenvalues of the residual covariance matrix.

Usage

```
mlmtst(fit, X, subset = NULL, tol = 0.001)
```

Arguments

fit	multivariate fit obtained by lm.
X	design matrix obtained by model.matrix.
subset	subset of predictors for which summary statistics will be reported. Note that this is different from the "subset" argument in lm.
tol	e[e/sum(e) > tol], where e is the vector of eigenvalues of the residual covariance matrix. Required to prevent long running times of algorithm AS 204. Default is 0.001 to ensure minimal loss of accuracy.

Details

Different types of sums of squares (i.e. "I", "II" and "III") are available.

Value

A list containing:

SS	sums of squares for all predictors (and residuals).
df	degrees of freedom for all predictors (and residuals).
f.tilde	pseudo-F statistics for all predictors.
r2	partial R-squared for all predictors.
e	eigenvalues of the residual covariance matrix.

Author(s)

Diego Garrido-Martín

See Also

AS204

`norm_parameters` *Settings for parameters of epi_preprocess function*

Description

`norm_parameters` function allows the user to set the values of the parameters to compute the functions `epi_preprocess`.

Usage

```
norm_parameters(
  illumina = list(bg.correct = TRUE, normalize = c("controls", "no"), reference = 1),
  quantile = list(fixOutliers = TRUE, removeBadSamples = FALSE, badSampleCutoff = 10.5,
    quantileNormalize = TRUE, stratified = TRUE, mergeManifest = FALSE, sex = NULL),
  noob = list(offset = 15, dyeCorr = TRUE, dyeMethod = c("single", "reference")),
  funnorm = list(nPCs = 2, sex = NULL, bgCorr = TRUE, dyeCorr = TRUE, keepCN = FALSE)
)
```

Arguments

`illumina`, `quantile`, `noob`, `funnorm`
: preprocess method selected in the function epi_preprocess.

`bg.correct`
: logical. If TRUE background correction will be performed in `"illumina"` method. Default is TRUE.

`normalize`
: logical. If TRUE control normalization will be performed in `"illumina"` method.

`reference`
: numeric. The reference array for control normalization in `"illumina"` method.

`fixOutliers`
: logical. If TRUE low outlier Meth and Unmeth signals will be fixed in `"quantile"` method. Default is TRUE.

`removeBadSamples`
: logical. If TRUE bad samples will be removed.

`badSampleCutoff`
: a numeric specifying the cutoff to label samples as 'bad' in `"quantile"` method. Default is 10.5.

`quantileNormalize`
: logical. If TRUE quantile normalization will be performed in `"quantile"` method. Default is TRUE.

`stratified`
: logical. If TRUE quantile normalization will be performed within region strata in `"quantile"` method. Default is TRUE.

`mergeManifest`
: logical. If TRUE the information in the associated manifest package will be merged into the output object in `"quantile"` method. Default is FALSE.

`offset`
: a numeric specifying an offset for the normexp background correction in `"noob"` method. Default is 15.

dyeCorr	logial. Dye correction will be done in "noob" and "funnorm" methods. Default is TRUE.
dyeMethod	specify the dye bias correction to be done, single sample approach or a reference array in "noob" method.
nPCs	numeric specifying the number of principal components from the control probes PCA in "funnorm" method. Default is 2.
sex	an optional numeric vector containing the sex of the samples in "quantile" and "funnorm" methods.
bgCorr	logical. If TRUE NOOB background correction will be done prior to functional normalization. in "funnorm" method. Default is TRUE.
keepCN	logical. If TRUE copy number estimates will be kept in "funnorm" method. Default is FALSE.

Details

Invoking epi_parameters() with no arguments returns a list with the default values for each normalization parameter.

Value

the function returns a list of all set parameters for each normalization method used in epi_peprocess.

Examples

```
#Default set of parameters
norm_parameters()
#change p value for manova method
norm_parameters(illumina = list("bg.correct" = FALSE))
```

---

p.asympt *Asymptotic P-values*

Description

Computes asymptotic P-values given the numerator of the pseudo-F statistic, its degrees of freedom and the eigenvalues of the residual covariance matrix.

Usage

```
p.asympt(ss, df, lambda, eps = 1e-14, eps.updt = 2, eps.stop = 1e-10)
```

Arguments

ss	numerator of the pseudo-F statistic.
df	degrees of freedom of the numerator of the pseudo-F statistic.
lambda	eigenvalues of the residual covariance matrix.
eps	the desired level of accuracy.
eps.updt	factor by which eps is updated to retry execution of algorithm AS 204 when it fails with fault indicator 4, 5 or 9.
eps.stop	if eps > eps.stop, execution of algorithm AS 204 is not retried and the function raises an error. Default is 1e-10.

Value

A vector containing the P-value and the level of accuracy.

Author(s)

Diego Garrido-Martín

See Also

AS204

`plot_epimutations` *Plot a given epimutation and locate it along the genome*

Description

This function plots a given epimutation and UCSC annotations for the specified genomic region.

Usage

```
plot_epimutations(
  dmr,
  methy,
  genome = "hg19",
  genes_annot = FALSE,
  regulation = FALSE,
  from = NULL,
  to = NULL
)
```

Arguments

`dmr`	epimutation obtained as a result of epimutations function.
`methy`	a GenomicRatioSet object containing the information of control and case samples used for the analysis in the epimutations function. See the constructor function GenomicRatioSet, makeGenomicRatioSetFromMatrix.
`genome`	a character string specifying the genome of reference. It can be set as `"hg38"`, `"hg19"` and `"hg18"`. The default is `"hg19"`.
`genes_annot`	a boolean. If TRUE gene annotations are plotted. Default is FALSE.
`regulation`	a boolean. If TRUE UCSC annotations for CpG Islands, H3K27Ac, H3K4Me3 and H3K27Me3 are plotted. The default is FALSE. The running process when `regulation` is TRUE can take several minutes.
`from, to`	scalar, specifying the range of genomic coordinates for the plot of gene annotation region. If `NULL` the plotting ranges are derived from the individual track. Note that `from` cannot be larger than `to`.

Details

The tracks are plotted vertically. Each track is separated by different background colour and a section title. The colours and titles are preset and cannot be set by the user.

Note that if you want to see the UCSC annotations maybe you need to take a bigger genomic region.

Value

The function returns a plot divided in two parts:

- ggplot graph including the individual with the epimutation in red, the control samples in dashed black lines and population mean in blue. Grey shaded regions indicate 1, 1.5 and 2 standard deviations from the mean of the distribution.
- UCSC gene annotations for the specified genomic region (if `genes == TRUE`)
- UCSC annotations for CpG Islands, H3K27Ac, H3K4Me3 and H3K27Me3 (if `regulation == TRUE`)

Examples

```
data(GRset)
data(res.epi.manova)
plot_epimutations(res.epi.manova[1,], GRset)
```

`process_ENSEMBL_results`

Process data from ENSEMBL

Description

Process data from ENSEMBL to combine results from the same regulatory elements in a unique record.

Usage

```
process_ENSEMBL_results(ensembl_res)
```

Arguments

`ensembl_res`	Results from `biomaRt::getBM`

Value

`data.frame` of one row after collapsing the input ENSEMBL regulatory regions

`res.epi.manova` *res.epi.manova*

Description

A data frame containing the results of `epimutations` function using `"manova"` methods for `GRset` dataset. For more information see the example of `epimutations` function.

Usage

```
data(res.epi.manova)
```

Format

A data frame with 16 variables and 6 epimutations.

Value

A data frame with 16 variables and 6 epimutations.

Examples

```
data(res.epi.manova)
```

`res_iForest` *Creates a data frame containing the results obtained from Isolation Forest*

Description

Creates a data frame containing the genomic regions, statistics and direction for the DMRs.

Usage

```
res_iForest(bump, sts, outlier_score_cutoff)
```

Arguments

`bump`	a DMR obtained from bumphunter (i.e. a row from bumphunter method result).
`sts`	the outlier score from epi_iForest function results.
`outlier_score_cutoff`	numeric specifying the outlier score cut off

Value

The function returns a data frame containing the following information for each DMR:

- genomic ranges
- DMR base pairs
- number and name of CpGs in DMR
- statistics:
  - Outlier score
  - Outlier significance
  - Outlier direction
- Sample name

For more information about the output see epimutations.

`res_mahdist` *Creates a data frame containing the results obtained from Robust Mahalanobis distance*

Description

Creates a data frame containing the genomic regions, statistics and direction for the DMRs.

Usage

```
res_mahdist(case, bump, outliers)
```

Arguments

`case`	a character string specifying the case sample name.
`bump`	a DMR obtained from bumphunter (i.e. a row from bumphunter method result).
`outliers`	the robust distance computed by epi_mahdist function results.

Value

The function returns a data frame containing the following information for each DMR:

- genomic ranges
- DMR base pairs
- number and name of CpGs in DMR
- statistics:
  - Outlier score
  - Outlier significance
  - Outlier direction
- Sample name

For more information about the output see epimutations.

res_manova *Creates a data frame containing the results obtained from MANOVA*

Description

Creates a data frame containing the genomic regions, statistics and direction for the DMRs.

Usage

```
res_manova(bump, sts)
```

Arguments

bump	a DMR obtained from bumphunter (i.e. a row from bumphunter method result).
sts	F statistic, Pillai and P value from epi_manova function results.

Value

The function returns a data frame containing the following information for each DMR:

- genomic ranges
- DMR base pairs
- number and name of CpGs in DMR
- statistics:
  - Outlier score
  - Outlier significance
  - Outlier direction
- Sample name

For more information about the output see epimutations.

res_mlm *Creates a data frame containing the results obtained from MLM*

Description

Creates a data frame containing the genomic regions, statistics and direction for the DMRs.

Usage

```
res_mlm(bump, sts)
```

Arguments

bump	a DMR obtained from bumphunter (i.e. a row from bumphunter method result).
sts	the F statistic, R2 test statistic and Pillai obtained as a result of epi_mlm function.

Value

The function returns a data frame containing the following information for each DMR:

- genomic ranges
- DMR base pairs
- number and name of CpGs in DMR
- statistics:
  - Outlier score
  - Outlier significance
  - Outlier direction
- Sample name

For more information about the output see epimutations.

UCSC_annotation *UCSC gene annotations*

Description

UCSC gene annotations for a given genome assembly.

Usage

```
UCSC_annotation(genome = "hg19")
```

Arguments

genome	genome asambly. Can be set as: 'hg38', 'hg19' and 'hg18'.

Value

The function returns gene annotations for the specified genome assembly.

UCSC_regulation *UCSC annotation*

Description

UCSC annotations for CpG Islands, H3K27Ac and H3K4Me3 for a given genome assembly and genomic coordinates.

Usage

```
UCSC_regulation(genome, chr, from, to)
```

Arguments

genome	genome asambly. Can be set as: 'hg38', 'hg19' and 'hg18'.
chr	a character string containing the sequence names to be analysed.
from, to	scalar, specifying the range of genomic coordinates. Note that from cannot be larger than to.

Value

UCSC_regulation returns a list containing CpG Islands, H3K27Ac and H3K4Me3 tacks.

Index