Package 'meffil'

March 17, 2024

Version 1.3.7

Title Efficient algorithms for DNA methylation

Description Tools for normalizing and analyzing the Infinium HumanMethylation450 BeadChip.

Author Matthew Suderman <matthewsuderman@hotmail.com> [cre, aut],
Gibran Hemani [cre, aut],
Josine Min [cre, aut]

Maintainer Matthew Suderman <matthewsuderman@hotmail.com>

Depends illuminaio,
MASS,
lmtest,
sandwich,
limma,
sva,
ggplot2,
plyr,
reshape2,
knitr,
Cairo,
gridExtra,
markdown,
matrixStats,
multcomp,
lme4,
parallel,
fastICA,
DNAcopy,
quadprog,
statmod,
gdsfmt,
SmartSVA,
preprocessCore

Suggests betareg,
minfi,
CopyNumber450kData

License Artistic-2.0

URL https://github.com/perishky/meffil

LazyData yes

biocViews DNAMethylation, Microarray, TwoChannel, DataImport, Preprocessing, QualityControl

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

R topics documented:

guess.batch.vars *Guess which columns in sample sheet are batch variables*

Description

Guess which columns in sample sheet are batch variables

Usage

```
guess.batch.vars(norm.objects)
```

Arguments

norm.objects	Output from meffil.normalize.quantiles

Value

Array of variable names

meffil.add.cell.type.reference *Create a cell type reference object*

Description

Create a cell type reference object for estimating cell counts with the Infinium HumanMethylation450 BeadChip.

Usage

```
meffil.add.cell.type.reference(
  name,
  M,
  U,
  cell.types,
  chip = NA,
  featureset = chip,
  number.sites = 50,
  specific.sites = NULL,
  number.quantiles = 500,
  subsets = NULL,
  object = NULL,
  description = NULL,
  verbose = F
)
```

Arguments

name	Character string providing the name of the reference.
M	Matrix of methylated probe intensities (rows=CpG sites, columns=samples).
U	Matrix of unmethylatd probe intensities (rows=CpG sites, columns=samples).
cell.types	Vector of cell type names corresponding to sample basenames.
chip	Name returned by meffil.list.chips() (Default: NA).
featureset	Name returned by meffil.list.featuresets() (Default: chip).
number.sites	Number of probes to characterise cell type methylation (Default: 50). For each cell type, this number of probes with greater methylation than other cell types and the same number with lesser methylation than the other cell types will be included.
specific.sites	If not null (default), then number.sites is ignored and the supplied site identifiers are used to differentiate between cell types instead of those maximally different between the cell types within the reference.
number.quantiles	Length of numeric sequence to specify probe intensity distributions (Default: 500).
object	Cell type reference previously created by this function. If not NULL, then this reference is added and all other function arguments are ignored (Default: NULL).
description	Text description of the reference (Default: NULL).
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

A list specifying a cell type reference object that can be used by meffil.estimate.cell.counts() to estimate cell counts in another dataset. The object is a list containing:

- beta The normalized methylation values of sites differentially methylated between cell types.
- quantiles The average quantiles of methylated and unmethylated signals of probe sets defined by subsets (see below). e.g. quantiles[[name]]$M provides the quantiles (number.quantiles quantiles) of the probes specified by subsets[[name]].
- subsets Probes on the microarray partitioned by relationship to CpG islands, either in an island, in a shore or far from an island.

---

meffil.add.chip *Add a new chip for analysis.*

Description

Add a new chip for analysis.

Usage

```
meffil.add.chip(name, manifest)
```

Arguments

name	Name of the new chip.
manifest	A data frame obtained by loading the Illumina manifest into R.

Value

Assuming that manifest contains a satisfactory set of columns, a new feature set and a new chip is made available. Thus, name will be added to the vectors returned by meffil.list.featuresets() and meffil.list.chips().

The manifest must contain the following columns:

- "IlmnID"character
- "Name"character
- "AddressA_ID"character
- "AddressB_ID"character
- "Infinium_Design_Type"values "I","II" or ""
- "CHR"values "0"-"22", "M", "X" or "Y"
- "MAPINFO"integer
- "AlleleA_ProbeSeq"character
- UCSC_RefGene_Namecharacter
- UCSC_RefGene_Accessioncharacter
- UCSC_RefGene_Groupcharacter
- "UCSC_CpG_Islands_Name"character
- "Relation_to_UCSC_CpG_Island"character
- "snp.exclude"logical

---

meffil.add.cnv.reference

Create a copy number reference object

Description

Create a copy number reference object for estimating copy number variation with the Infinium HumanMethylation450 BeadChip.

Usage

```
meffil.add.cnv.reference(
  name,
  M,
  U,
  chip = NA,
  featureset = chip,
  object = NULL,
  verbose = T
)
```

Arguments

name	Character string providing the name of the reference.
M	Matrix of methylated probe intensities (rows=CpG sites, columns=samples).
U	Matrix of unmethylatd probe intensities (rows=CpG sites, columns=samples).
chip	Name returned by meffil.list.chips() (Default: NA).
featureset	Name returned by meffil.list.featuresets() (Default: chip).
object	A previously created copy number reference object created by this function. If not NULL, then this reference is added with the given name and all other function arguments are ignored (Default: NULL).
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

A list specifying a copy number reference object that can be used by meffil.calculate.cnv() to estimate copy number variation in another dataset.

---

meffil.add.copynumber450k.references

Create copy number references from CopyNumber450kData

Description

Two copy number references are created using data from the Bioconductor CopyNumber450kData R package. Reference "copynumber450k" is created using the "450k" feature set, and reference "copynumber450k-common" is created using the "common" feature set so it can be used with datasets with mixed 450K and EPIC chips.

Usage

```
meffil.add.copynumber450k.references(verbose = T)
```

---

meffil.add.featureset *Add a feature set.*

Description

Add a feature set.

Usage

```
meffil.add.featureset(name, features)
```

Arguments

name	Name of the new feature set.
features	A data frame listing and describing all features.

Value

Assuming that `features` contains a satisfactory set of columns, a new feature set is made available. Thus, `name` will be added to the vector returned by `meffil.list.featuresets()`.

The `features` data frame must contain the following columns:

- "name"character
- "target"character
- "type"values "i","ii" or "control"
- "chromosome"values "chr0"-"chr22", "chrM", "chrX" or "chrY"
- "position"integer
- "gene.symbol"character,
- "gene.accession"character,
- "gene.region"character,
- "cpg.island.name"character
- "relation.to.island"character
- "snp.exclude"logical

`meffil.all.features` *Feature information for all Illumina Beadchips*

Description

Feature information for all Illumina Beadchips

Usage

```
meffil.all.features()
```

Value

A data frame listing all features.

`meffil.autosomal.subset` *Restrict to subset of features targetting autosomal CpG sites*

Description

Restrict to subset of features targetting autosomal CpG sites

Usage

```
meffil.autosomal.subset(features)
```

Arguments

`features`	A vector of feature names.

Value

Subset of the input targetting autosomal CpG sites

---

meffil.basenames *IDAT file basenames*

Description

List IDAT file basenames in a given directory.

Usage

```
meffil.basenames(path, recursive = FALSE)
```

Arguments

path	Directory containing the IDAT files.
recursive	If TRUE, search for IDAT files in subdirectories as well (Default: FALSE).

Value

Character vector of IDAT file basenames (i.e. filenames with "_Grn.idat" and "_Red.idat" removed). In other words, each identifies the Cy5 and Cy3 output files corresponding to a single microarray.

---

meffil.calculate.cnv *Calculate CNVs from IDAT files*

Description

Based on the algorithm developed in R/CopyNumber450k bioconductor package

Usage

```
meffil.calculate.cnv(
  samplesheet,
  cnv.reference,
  chip = NA,
  verbose = FALSE,
  ...
)
```

Arguments

samplesheet	Output from meffil.create.samplesheet
cnv.reference	Name returned by meffil.list.cnv.references().
chip	Name returned by meffil.list.chips() (Default: NA).
verbose	Default = FALSE
...	Extra parameters to be passed to DNAcopy for segmentation. See details.

Details

The following default values are being used:

- trim = 0.1
- min.width = 5
- nperm = 10000
- alpha = 0.001
- undo.splits = "sdundo"
- undo.SD = 2

Value

Dataframe of segmented results

`meffil.cell.count.estimates`
Cell count estimates

Description

Cell count estimates

Usage

```
meffil.cell.count.estimates(qc.objects)
```

Arguments

`qc.objects`	List of objects obtained from `meffil.qc()` or `meffil.create.qc.object()`.

`meffil.cell.count.qc.plots`
Cell count estimate quality plot

Description

Cell count estimate quality plot

Usage

```
meffil.cell.count.qc.plots(count.objects)
```

Arguments

`count.objects`	A list of objects each obtained from `meffil.estimate.cell.counts()`.

Value

Two ggplot2 boxplot objects:

- `betas` Contains one box per sample or reference cell type representing the distribution of methylation levels for the CpG sites used to estimate cell counts.
- `counts` Contains one box per reference cell type representing the distribution of cell count estimates across the samples.

`meffil.cell.type.specific.methylation`
Reduce methylation profiles to most cell-type specific sites

Description

Reduce methylation profiles to most cell-type specific sites

Usage

```
meffil.cell.type.specific.methylation(
  beta,
  cell.types,
  number.sites = 50,
  verbose = F
)
```

Arguments

`beta`	Numeric matrix (values = 0..1; rows = CpG sites; columns = samples).
`cell.types`	Name of cell type for each column of beta.
`number.sites`	For each cell type, the number of sites less methylated and the number more methylated than other cell types to include in the reduced methylation profiles.

Value

Numeric matrix (values = 0..1; rows = CpG sites; columns = cell types) with `number.sites` CpG sites per cell type more methylated than other cell types and the same number less methylated. Values are the mean CpG site methylation levels of all original samples of the same cell type.#

meffil.cnv.matrix *Create matrix of CNV values*

Description

Create matrix of CNV values

Usage

```
meffil.cnv.matrix(cnv, featureset = "450k")
```

Arguments

cnv	Output from meffil.calculate.cnv().
featureset	Name from meffil.list.featuresets() (Default: "450k").

Value

Matrix of ncpg x nsample

meffil.collapse.dups *Collapse duplicate probes*

Description

Collapse duplicated probes by replacing them with a summary.

Usage

```
meffil.collapse.dups(beta, dup.fun = function(x) median(x, na.rm = T))
```

Arguments

beta	Methylation matrix returned by meffil.normalize.samples().
dup.fun	Function to collapse duplicate probes (Default: median).

Value

The input matrix with duplicated probes (i.e. row names identical after stripping everything after the "_" character) replaced by summaries defined by dup.fun.

meffil.control.matrix *Infinium HumanMethylation450 BeadChip control matrix*

Description

Matrix containing control probe intensities from the Infinium HumanMethylation450 BeadChip.

Usage

```
meffil.control.matrix(
  qc.objects,
  normalize = F,
  fixed.effects = NULL,
  random.effects = NULL
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
normalize	If TRUE, then control matrix is scaled and specified fixed and random effects removed from the matrix. Otherwise, the raw control matrix is returned. (Default: FALSE).
fixed.effects	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).
random.effects	Names of columns in samplesheet that should be included as random effects (Default: NULL).

Value

Matrix with one row per object consisting of control probe intensities and summaries.

meffil.create.qc.object *Quality control object*

Description

Create a quality control object for a given Infinium HumanMethylation450 BeadChip.

Usage

```
meffil.create.qc.object(
  samplesheet.row,
  number.quantiles = 500,
  dye.intensity = 5000,
  verbose = F,
  detection.threshold = 0.01,
  bead.threshold = 3,
  sex.cutoff = -2,
```

```
  chip = NA,
  featureset = chip,
  cell.type.reference = NA
)
```

Arguments

samplesheet.row	Row from the data frame containing IDAT file and sample info (see meffil.read.samplesheet or meffil.create.samplesheet).
number.quantiles	Number of quantiles to compute for probe subset (Default: 500).
dye.intensity	Reference intensity for scaling each color channel (Default: 5000).
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).
detection.threshold	Default value = 0.01. All probes above this detection threshold detected.
bead.threshold	Default value = 3. All probes with less than this number of beads detected.
sex.cutoff	Sex prediction cutoff. Default value = -2.
chip	Name returned by meffil.list.chips() (Default: NA).
featureset	Name returned by meffil.list.featuresets() (Default: chip).
cell.type.reference	Character string name of the cell type reference to use for estimating cell counts. Estimates are not generated if set to NA (default). See meffil.list.cell.type.references() for a list of available references. New references can be created using meffil.add.cell.type.refer

Value

List containing control probe information, probe summaries and quantiles. We call this a "QC object".

---

meffil.create.samplesheet

Create sample sheet if an Illumina one isn't available

Description

If necessary generates two columns necessary for some functions: Sample_Name and Sex

Usage

```
meffil.create.samplesheet(
  path,
  basenames = meffil.basenames(path, recursive),
  recursive = FALSE,
  delim = "_"
)
```

Arguments

basenames	Output from meffil.basenames
delim	Optional delim character to separate Sample_Name into multiple columns. Default: "_"

Value

Sample sheet data frame

---

meffil.design.matrix *Infinium HumanMethylation450 BeadChip normalization design matrix*

Description

Design matrix derived by applying principal components analysis to control probes.

Usage

```
meffil.design.matrix(
  qc.objects,
  number.pcs,
  fixed.effects = NULL,
  random.effects = NULL
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
number.pcs	Number of principal components to include in the design matrix (Default: all).
fixed.effects	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).
random.effects	Names of columns in samplesheet that should be included as random effects (Default: NULL).

Value

Design matrix with one column for each of the first number.pcs prinicipal components.

`meffil.estimate.cell.counts` *Estimate cell counts from a reference*

Description

Estimate cell type ratios from methylation profiles of purified cell populations (Infinium HumanMethylation450 BeadChip) using the the Houseman algorithm (PMID 22568884).

Usage

```
meffil.estimate.cell.counts(qc.object, cell.type.reference, verbose = T)
```

Arguments

`cell.type.reference`	Character string name of the cell type reference to use for estimating cell counts. See `meffil.list.cell.type.references()` for a list of available references. New references can be created using `meffil.add.cell.type.reference()`.
`verbose`	If `TRUE`, then status messages are printed during execution (Default: `FALSE`).
`object`	An object created by `meffil.create.qc.object()`.

Value

A list:

- `counts` Cell count estimates.
- `beta` Normalized methylation levels of sites used to differentiate
- `reference` Name of the cell type reference used. between reference cell types.

Results should be nearly identical to `estimateCellCounts()`.

`meffil.estimate.cell.counts.from.betas` *Estimate cell counts for a methylation matrix from a reference*

Description

Estimate cell counts for a methylation matrix from a reference

Usage

```
meffil.estimate.cell.counts.from.betas(beta, cell.type.reference, verbose = F)
```

Arguments

beta	Matrix of methylation levels (rows = CpG sites, columns = subjects).
cell.type.reference	Character string name of the cell type reference to use for estimating cell counts. See meffil.list.cell.type.references() for a list of available references. New references can be created using meffil.add.cell.type.reference().
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

A matrix of cell count estimates.

Results should be nearly identical to estimateCellCounts().

---

meffil.ewas *Epigenome-wide association study*

Description

Test association with each CpG site.

Usage

```
meffil.ewas(
  beta,
  variable,
  covariates = NULL,
  batch = NULL,
  weights = NULL,
  sites = NULL,
  samples = NULL,
  cell.counts = NULL,
  isva = F,
  sva = T,
  smartsva = F,
  smartsva.alpha = 0.5,
  n.sv = NULL,
  winsorize.pct = 0.05,
  robust = FALSE,
  rlm = FALSE,
  outlier.iqr.factor = NA,
  most.variable = 50000,
  featureset = NA,
  random.seed = 20161123,
  lmfit.safer = F,
  verbose = F
)
```

Arguments

beta	Methylation levels matrix, one row per CpG site, one column per sample or the filename of GDS (Genomic Data Structure) output from meffil.normalize.samples.
variable	Independent variable vector.
covariates	Covariates data frame to include in regression model, one row per sample, one column per covariate (Default: NULL).
batch	Batch vector to be included as a random effect (Default: NULL). Ignored if beta is a GDS filename.
weights	Non-negative observation weights. Can be a numeric matrix of individual weights of same dimension as beta, or a numeric vector of weights with length ncol(beta), or a numeric vector of weights with length nrow(beta).
sites	Restrict the EWAS to the given CpG sites – must match row names of beta (Default: NULL).
samples	Restrict the EWAS to the given samples – must match column names of beta (Default: NULL).
cell.counts	Proportion of cell counts for one cell type in cases where the samples are mainly composed of two cell types (e.g. saliva) (Default: NULL). Ignored if beta is a GDS filename.
isva	Apply Independent Surrogate Variable Analysis (ISVA) to the methylation levels and include the resulting variables as covariates in a regression model (Default: FALSE).
sva	Apply Surrogate Variable Analysis (SVA) to the methylation levels and covariates and include the resulting variables as covariates in a regression model (Default: TRUE).
smartsva	Apply the SmartSVA algorithm to the methylation levels and include the resulting variables as covariates in a regression model (Default: FALSE).
smartsva.alpha	alpha argument to SmartSVA providing the initial point for optimization. Smaller values reduce the number of iterations needed to reach convergence. Setting this 1 will produce exactly the outputs as SVA. (Default: 0.5).
n.sv	Number of surrogate variables to calculate (Default: NULL).
winsorize.pct	Apply all regression models to methylation levels winsorized to the given level. Set to NA to avoid winsorizing (Default: 0.05).
robust	Test associations with the 'robust' option when limma::eBayes is called (Default: TRUE). Ignored if beta is a GDS filename.
rlm	If beta is a matrix, then test associations with the 'robust' option when limma:lmFit is called. If beta is a GDS filename, then test associations using robust regression using MASS::rlm and calculate statistical significance using lmtest::coeftest with vcov=sandwich::vcovHC(fit, type="HC0") (Default: FALSE).
outlier.iqr.factor	For each CpG site, prior to fitting regression models, set methylation levels less than Q1 - outlier.iqr.factor * IQR or more than Q3 + outlier.iqr.factor * IQR to NA. Here IQR is the inter-quartile range of the methylation levels at the CpG site, i.e. Q3-Q1. Set to NA to skip this step (Default: NA).
most.variable	Apply (Independent) Surrogate Variable Analysis to the given most variable CpG sites (Default: 50000).
featureset	No longer used (Default: NA).
verbose	Set to TRUE if status updates to be printed (Default: FALSE).

meffil.ewas.bedgraph *Save EWAS effect estimates to bedgraph file*

Description

Saves EWAS effect estimates to a bedgraph file for viewing on a genome browser. More file format details can be found here: https://genome.ucsc.edu/goldenPath/help/bedgraph.html

Usage

```
meffil.ewas.bedgraph(
  ewas.object,
  filename,
  analysis,
  name,
  description,
  header
)
```

Arguments

ewas.object	Object returned by meffil.ewas().
filename	Filename for output, typically with a 'bed' extension.
analysis	The particular EWAS analysis from which to obtain summary statistics. This should be one of names(ewas.object$analyses).
name	Text name to be included in the bedgraph header.
description	Text description to be included in the bedgraph header.
header	Bedgraph header. The default header uses the name and description provided.

meffil.ewas.covariate.associations *Describe associations between EWAS covariates and the variable of interest.*

Description

Describe associations between EWAS covariates and the variable of interest.

Usage

```
meffil.ewas.covariate.associations(ewas.object)
```

Arguments

ewas.object	Output of meffil.ewas().

Value

A data frame with one or more rows for each covariate.

If both the variable of interest and covariate are continuous or ordinal, then the covariate uses one row showing the name, mean and standard deviation of the covariate following the significance of the association between the covariate and the variable of interest.

If the covariate is categorical, then there is additionally one row for each level showing the mean and standard deviation of the variable of interest for samples at that covariate level.

If the variable of interest is categorical but the covariate is not, then there is one row for each variable level showing the mean and standard deviation of the covariate at the given level.

If both the variable of interest and covariate are categorical, then mean is replaced with the number of samples at each pair of variable/categorical levels and standard deviation with the percentage. P-values indicate the significance of association using Fisher's exact test.

`meffil.ewas.cpg.plot` *Scatter plots for a CpG site in an EWAS*

Description

Scatter plots for a CpG site in an EWAS

Usage

```
meffil.ewas.cpg.plot(ewas.object, cpg, beta, title = cpg)
```

Arguments

`ewas.object`	Return object from `meffil.ewas()`.
`cpg`	CpG site to plot.
`beta`	Matrix of methylation levels used to create the `ewas.object`.
`title`	Title of the plot (Default: `cpg`).
`ggplot`	object showing the scatterplots of DNA methylation vs the variable of interest in the EWAS. Each plot corresponds to a covariate set. Methylation levels are in fact residuals from fitting a model with DNA methylation and the covariates.

`meffil.ewas.manhattan.plot`
Manhattan plot for EWAS

Description

Manhattan plot for EWAS

Usage

```
meffil.ewas.manhattan.plot(
  ewas.object,
  sig.threshold = 1e-07,
  title = "Manhattan plot"
)
```

Arguments

ewas.object	Return object from meffil.ewas().
sig.threshold	P-value threshold for significance (Default: 1e-7).
title	Title for the plot (Default: "Manhattan plot").

Value

ggplot showing the Manhattan plot.

---

meffil.ewas.old *Epigenome-wide association study (OLD VERSION RETAINED FOR COMPARISON)*

Description

Test association with each CpG site.

Usage

```
meffil.ewas.old(
  beta,
  variable,
  covariates = NULL,
  batch = NULL,
  weights = NULL,
  cell.counts = NULL,
  isva = T,
  sva = T,
  smartsva = F,
  n.sv = NULL,
  isva0 = F,
  isva1 = F,
  winsorize.pct = 0.05,
  robust = TRUE,
  rlm = FALSE,
  outlier.iqr.factor = NA,
  most.variable = min(nrow(beta), 50000),
  featureset = NA,
  random.seed = 20161123,
  lmfit.safer = F,
  verbose = F
)
```

Arguments

beta	Methylation levels matrix, one row per CpG site, one column per sample.
variable	Independent variable vector.
covariates	Covariates data frame to include in regression model, one row per sample, one column per covariate (Default: NULL).

batch	Batch vector to be included as a random effect (Default: NULL).
weights	Non-negative observation weights. Can be a numeric matrix of individual weights of same dimension as beta, or a numeric vector of weights with length ncol(beta), or a numeric vector of weights with length nrow(beta).
cell.counts	Proportion of cell counts for one cell type in cases where the samples are mainly composed of two cell types (e.g. saliva) (Default: NULL).
isva	Apply Independent Surrogate Variable Analysis (ISVA) to the methylation levels and include the resulting variables as covariates in a regression model (Default: TRUE).
sva	Apply Surrogate Variable Analysis (SVA) to the methylation levels and covariates and include the resulting variables as covariates in a regression model (Default: TRUE).
smartsva	Apply the SmartSVA algorithm to the methylation levels and include the resulting variables as covariates in a regression model (Default: FALSE).
n.sv	Number of surrogate variables to calculate (Default: NULL).
winsorize.pct	Apply all regression models to methylation levels winsorized to the given level. Set to NA to avoid winsorizing (Default: 0.05).
robust	Test associations with the 'robust' option when limma::eBayes is called (Default: TRUE).
rlm	Test assocaitions with the 'robust' option when limma:lmFit is called (Default: FALSE).
outlier.iqr.factor	For each CpG site, prior to fitting regression models, set methylation levels less than Q1 - outlier.iqr.factor * IQR or more than Q3 + outlier.iqr.factor * IQR to NA. Here IQR is the inter-quartile range of the methylation levels at the CpG site, i.e. Q3-Q1. Set to NA to skip this step (Default: NA).
most.variable	Apply (Independent) Surrogate Variable Analysis to the given most variable CpG sites (Default: 50000).
featureset	Name from meffil.list.featuresets() (Default: NA).
verbose	Set to TRUE if status updates to be printed (Default: FALSE).

`meffil.ewas.parameters`

Specify parameters for QC

Description

Specify parameters for QC

Usage

```
meffil.ewas.parameters(
  sig.threshold = NA,
  max.plots = 10,
  model = "none",
  qq.inflation.method = "median"
)
```

Arguments

sig.threshold	P-value threshold for significance (Default: NA). If NA, then threshold used will be 0.05 divided by the number of tests/probes.
max.plots	Maximum number of plots to generate (Default: 10).
model	Model to use for selecting associations: "none" (no covariates), "all" (all covariates), "isva" (independent surrogate variables), and "sva" (surrogate variables) (Default: "none").
qq.inflation.method	Method for calculating genomic inflation lambda. Valid values are "median", "regression" or "robust" (Default: "median").

Value

List of parameter values

---

meffil.ewas.qq.plot *QQ plot for EWAS*

Description

QQ plot for EWAS

Usage

```
meffil.ewas.qq.plot(
  ewas.object,
  sig.threshold = 1e-07,
  sig.color = "red",
  title = "QQ plot",
  xlab = bquote(-log[10]("expected p-values")),
  ylab = bquote(-log[10]("observed p-values")),
  lambda.method = "median"
)
```

Arguments

ewas.object	Return object from meffil.ewas().
sig.threshold	P-value threshold for significance (Default: 1e-7).
sig.color	Color for points corresponding to significant tests (Default: "red").
title	Title for the plot (Default: "QQ plot").
xlab	Label for the x-axis (Default: -log_10(expected p-values)).
ylab	Label for the y-axis (Default: -log_10(observed p-values)).
lambda.method	Method for calculating genomic inflation lambda. Valid values are "median", "regression", or "robust" (Default: "median").

Value

List of ggplot for each analysis in ewas.object.

meffil.ewas.report *Generate EWAS report.*

Description

Generate HTML file that summarises the EWAS.

Usage

```
meffil.ewas.report(
  ewas.summary,
  output.file = "ewas-report.html",
  author = "Analyst",
  study = "Illumina methylation data",
  ...
)
```

Arguments

ewas.summary	Output from meffil.ewas.summary.
output.file	Default = "ewas-report.html". If the file extension is not .htm, .html, .HTM or .HTML then output will be in markdown format.
author	Default = "Analyst". Author name to be specified on report.
study	Default = "Illumina methylation data". Study name to be specified on report.
...	Arguments to be passed to knitr::knit

meffil.ewas.sample.characteristics *Describe EWAS samples using the variable of interest and covariates.*

Description

Describe EWAS samples using the variable of interest and covariates.

Usage

```
meffil.ewas.sample.characteristics(ewas.object)
```

Arguments

ewas.object	Output of meffil.ewas().

Value

A data frame with one row for each continuous or ordinal variable and one row for each level of each categorical variable. In the first case, each row provides the name, mean value and standard deviation of each variable. In the second case (categorical), each row provides the name of the variable level and the number of cases and percentage of cases at that level.

meffil.ewas.summary *Summarize EWAS results.*

Description

Generates variable and covariate summary tables, QQ plots, Manhattan plots, a list of associations, plots of the strongest associations and plots of selected CpG sites.

Usage

```
meffil.ewas.summary(
  ewas.object,
  beta,
  selected.cpg.sites = character(0),
  parameters = meffil.ewas.parameters(),
  verbose = T
)
```

Arguments

ewas.object	From meffil.ewas().
beta	Methylation levels used in the analysis, either a matrix with one row per CpG site and one column per sample or the filename of a GDS file (Genomic Data Structure).
selected.cpg.sites	Vector of CpG site names to plot (Default: character(0)).
parameters	Default = meffil.ewas.parameters(). List of parameter values. See meffil.ewas.parameters().

Value

List

meffil.extract.genotypes *Extract genotype data from PLINK .raw files for Illumina 450K SNPs*

Description

Extract genotype data from PLINK .raw files for Illumina 450K SNPs

Usage

```
meffil.extract.genotypes(filenames, verbose = F)
```

Arguments

filenames	A vector of filenames of PLINK .raw files from which to extract genotype data.

Value

Matrix with rows corresponding to SNPs, columns to samples and values equal to 0, 1 or 2 corresponding to genotypes.

Examples

```
R> writeLines(meffil.snp.names("450k"), con="snp-names.txt")
shell> plink --bfile dataset --extract snp-names.txt --recodeA --out genotypes.raw --noweb
R> filenames <- "genotypes.raw"
R> genotypes <- meffil.extract.genotypes(filenames)
```

`meffil.featureset` *Obtain a list of features in a feature set.*

Description

Obtain a list of features in a feature set.

Usage

```
meffil.featureset(featureset = "450k")
```

Arguments

`featureset`	Name returned by `meffil.list.featuresets()` (Default: "450k").

Value

A data frame with one row for each feature.

Examples

```
x <- meffil.featureset("450k")
```

`meffil.gds.apply` *Return a vector or list of values obtained by applying a function to the margins of a methylation or detection p-value matrix stored in a GDS file.*

Description

Return a vector or list of values obtained by applying a function to the margins of a methylation or detection p-value matrix stored in a GDS file.

Usage

```
meffil.gds.apply(
  gds.filename,
  bysite = T,
  type = c("list", "none", "integer", "double", "character", "logical", "raw"),
  FUN,
  sites = NULL,
  samples = NULL,
  ...
)
```

Arguments

gds.filename	Name of GDS file generated by meffil.normalize.samples()
bysite	If TRUE, then apply function to each CpG site (row), otherwise to each sample (column) (Default: TRUE).
type	returned value.
FUN	the function to be applied.
sites	Names of CpG sites to apply to, NULL means all sites (Default: NULL).
samples	Names of samples to apply to, NULL means all samples (Default: NULL).
...	

---

meffil.gds.detection.pvalues

Retrieve detection p-values from GDS file

Description

Retrieve detection p-values from GDS file

Usage

```
meffil.gds.detection.pvalues(gds.filename, sites = NULL, samples = NULL)
```

Arguments

gds.filename	Name of GDS file generated by meffil.save.detection.pvalues().
sites	Names of CpG sites to load, if NULL then load all (Default: NULL).
samples	Names of samples to load, if NULL then load all (Default: NULL).

Value

Matrix of methylation levels with rows corresponding to CpG sites and columns to samples. Rows restricted sites if not NULL, and columns restricted to samples if not NULL.

meffil.gds.dims *Retrieve methylation or detection p-value matrix row and column names*

Description

Retrieve methylation or detection p-value matrix row and column names

Usage

```
meffil.gds.dims(gds.filename)
```

Arguments

gds.filename	Name of GDS file generated by meffil.normalize.samples() or meffil.save.detection.pvalu

Value

A list of two vectors, the first providing the row names (CpG sites) and the second providing the column names (sample identifiers).

meffil.gds.methylation *Retrieve methylation levels from GDS file*

Description

Retrieve methylation levels from GDS file

Usage

```
meffil.gds.methylation(gds.filename, sites = NULL, samples = NULL)
```

Arguments

gds.filename	Name of GDS file generated by meffil.normalize.samples().
sites	Names of CpG sites to load, if NULL then load all (Default: NULL).
samples	Names of samples to load, if NULL then load all (Default: NULL).

Value

Matrix of methylation levels with rows corresponding to CpG sites and columns to samples. Rows restricted sites if not NULL, and columns restricted to samples if not NULL.

meffil.get.autosomal.sites

Get names of autosomal CpG sites in the feature set.

Description

Get names of autosomal CpG sites in the feature set.

Usage

```
meffil.get.autosomal.sites(featureset = "450k")
```

meffil.get.beta

Infinium HumanMethylation450 BeadChip methylation levels

Description

Compute beta values (methylation levels) from methylated/unmethylated signals

Usage

```
meffil.get.beta(M, U, pseudo = 100)
```

Arguments

M	Methylated signal matrix.
U	Unmethylated signal matrix.
pseudo	Value to add to the denominator to make the methylation estimate more stable.

Value

Matrix of 0..1 methylation level estimates. Equal to methylated/(methylated + unmethylated + pseudo).

meffil.get.features

Get a list of microarray features from a predefined feature set.

Description

Get a list of microarray features from a predefined feature set.

Usage

```
meffil.get.features(featureset = "450k")
```

Arguments

featureset	A name returned by meffil.list.featuresets() (Default: "450k").

Value

A data frame listing all features in the feature set.

`meffil.get.sites` *Get names of all CpG sites in the feature set.*

Description

Get names of all CpG sites in the feature set.

Usage

```
meffil.get.sites(featureset = "450k")
```

`meffil.get.typeii.sites` *Get names of CpG sites corresponding to Infinium Type II probes in the feature set.*

Description

Get names of CpG sites corresponding to Infinium Type II probes in the feature set.

Usage

```
meffil.get.typeii.sites(featureset = "450k")
```

`meffil.get.x.sites` *Get names of chromosome X CpG sites in the feature set.*

Description

Get names of chromosome X CpG sites in the feature set.

Usage

```
meffil.get.x.sites(featureset = "450k")
```

`meffil.get.y.sites` *Get names of chromosome Y CpG sites in the feature set.*

Description

Get names of chromosome Y CpG sites in the feature set.

Usage

```
meffil.get.y.sites(featureset = "450k")
```

meffil.handle.outliers

Handle outliers in a methylation matrix

Description

Handle outliers in a methylation matrix

Usage

```
meffil.handle.outliers(beta, winsorize.pct = 0.05, outlier.iqr.factor = NA)
```

Arguments

beta	Methylation matrix (rows=CpG sites, columns=samples, values=methylation levels).
winsorize.pct	Apply all regression models to methylation levels winsorized to the given level. Set to NA to avoid winsorizing (Default: 0.05).
outlier.iqr.factor	For each CpG site, prior to fitting regression models, set methylation levels less than Q1 - outlier.iqr.factor * IQR or more than Q3 + outlier.iqr.factor * IQR to NA. Here IQR is the inter-quartile range of the methylation levels at the CpG site, i.e. Q3-Q1. Set to NA to skip this step (Default: NA).

Value

beta after winsorizing and outliers set to NA.

meffil.list.cell.type.references

List of available cell type references

Description

List of available cell type references

Usage

```
meffil.list.cell.type.references()
```

Details

Names and description of available references:

- "andrews and bakulski cord blood" Derived from FlowSorted.CordBlood.450k
- "blood gse167998" Adult blood reference of Salas et al. Nat Comms 2022
- "blood gse35069" Adult blood reference of Reinius et al. PLoS One 2012
- "blood gse35069 chen" Adult blood reference of Reinius et al. PLoS One 2012 restricted to CpG sites of Table E2 in Chen et al. J Allergy Clin Immunol 2017

- "blood gse35069 complete" Adult blood reference of Reinius et al. PLoS One 2012 with neutrophils and eosinophils
- "blood idoloptimized" Derived from FlowSorted.Blood.450k
- "blood idoloptimized epic" Derived from FlowSorted.Blood.EPIC
- "combined cord blood" Derived from FlowSorted.CordBloodCombined.450k
- "cord blood gse68456" Cord blood reference of Goede et al. Clin Epigenetics 2015
- "gervin and lyle cord blood" Cord blood reference of Gervin et al. Epigenetics 2016
- "guintivano dlpfc" Derived from FlowSorted.DLPFC.450k
- "saliva gse48472" Saliva reference composed of buccal cell data from Slieker et al. Epigenetics Chromatin 2013 and (blood) immune cell data from Reinius et al. PLoS One 2012

Examples

```
## obtain a list of references
references <- meffil.list.cell.type.references()
## show descriptions for each
comment(references)
```

`meffil.list.chips` *List of microarrays formats available.*

Description

By default, there is '450k' and 'epic'. Additions can be made using `meffil.add.chip()`.

Usage

```
meffil.list.chips()
```

`meffil.list.cnv.references`
List of available copy number references

Description

List of available copy number references

Usage

```
meffil.list.cnv.references()
```

meffil.list.featuresets

List of feature sets available.

Description

Sets of features for individual platforms (e.g. "450k" for the Illumina HumanMethylation450 Beadchip) as well as for mixed platforms (e.g. "450k:epic:epic2" for combinations of Illumina HumanMethylation450, MethylationEPIC and MethylationEPICv2).

Usage

```
meffil.list.featuresets()
```

Details

In most cases, a feature corresponds to the two probes from which it's value is derived. Each CpG represented on the chip for example corresponds to a single feature derived from a probe measuring methylated signal and a second probe measuring unmethylated signal.

Each control feature corresponds to a unique control probe.

meffil.load.controls *Load control probes*

Description

Load control probes

Usage

```
meffil.load.controls(
  samplesheet,
  chip = NA,
  featureset = chip,
  verbose = F,
  ...
)
```

Arguments

samplesheet	Sample info (see meffil.read.samplesheet or meffil.create.samplesheet).
chip	Name returned by meffil.list.chips() (Default: NA).
featureset	Name returned by meffil.list.featuresets() (Default: chip).
verbose	(Default: FALSE).
...	Arguments to mclapply.

Value

List containing two elements: probes and values. The probes item is a data frame describing the control probes. The values item is a matrix providing the intensities of the control probes for each samples (rows=probes, columns=samples).

`meffil.load.detection.pvalues` *Load detection p-value matrix*

Description

Load detection p-value matrix

Usage

```
meffil.load.detection.pvalues(
  qc.objects,
  max.bytes = 2^30 - 1,
  verbose = F,
  ...
)
```

Arguments

`qc.objects`	A list of outputs from `meffil.create.qc.object()`.
`verbose`	If `TRUE`, then detailed status messages are printed during execution (Default: `FALSE`).
`...`	Arguments passed to `mclapply()`.

Value

Matrix of probe detection p-values.

`meffil.load.raw.data` *Load raw beta matrix*

Description

Load raw beta matrix

Usage

```
meffil.load.raw.data(
  qc.objects,
  pseudo = 100,
  just.beta = T,
  max.bytes = 2^30 - 1,
  verbose = F,
  ...
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
pseudo	Value to add to the denominator to make the methylation estimate more stable when calculating methylation levels (Default: 100).
just.beta	If TRUE, then return just the methylation levels; otherwise, return the methylated and unmethylated matrices (Default: TRUE).
verbose	If TRUE, then detailed status messages are printed during execution (Default: FALSE).
...	Arguments passed to mclapply().

Value

If just.beta == TRUE, the matrix of methylation levels between between 0 and 1 equal to methylated signal/(methylated + unmethylated signal + pseudo). Otherwise, a list containing two matrices, the methylated and unmethylated signals.

---

meffil.methylation.pcs

Compute principal components of a methylation matrix.

Description

Compute principal components of a methylation matrix.

Usage

```
meffil.methylation.pcs(
  beta,
  probe.range = 50000,
  sites = NULL,
  samples = NULL,
  autosomal = T,
  winsorize.pct = NA,
  outlier.iqr.factor = NA,
  full.obj = F,
  verbose = F
)
```

Arguments

beta	Output from meffil.normalize.samples(), either a matrix or a GDS filename.
probe.range	Default = 50000. How many probes to be used in calculating PCs.
sites	Subset of CpG sites to consider (row names of beta) (Default: NULL).
samples	Subset of samples to consider (column names of beta) (Default: NULL).
autosomal	If true, remove probes on sex chromosomes (Default: TRUE).
winsorize.pct	Apply to methylation levels winsorized to the given level. Set to NA to avoid winsorizing (Default: NA).

`outlier.iqr.factor`
: Apply to methylation after setting, for each CpG site, values less than `Q1 - outlier.iqr.factor * IQR` or more than `Q3 + outlier.iqr.factor * IQR` to NA. Here IQR is the inter-quartile range of the methylation levels at the CpG site, i.e. Q3-Q1. Set to NA to skip this step (Default: NA).

`full.obj`
: Default = FALSE. If true, then return the full `prcomp` object rather than just the PCs.

`verbose=T`
: Print progress messages?

Value

the principal components of `normalized.beta`.

`meffil.most.variable.cpgs`
Most variable CpG sites

Description

Returns the most variable CpG sites (rows) in the methylation matrix.

Usage

```
meffil.most.variable.cpgs(
  beta,
  n = 1000,
  sites = NULL,
  samples = NULL,
  autosomal = T,
  winsorize.pct = NA,
  outlier.iqr.factor = NA
)
```

Arguments

`beta`
: Output from `meffil.normalize.samples()`, either a matrix or a GDS filename.

`n`
: Number of CpG sites to return.

`sites`
: Subset of CpG sites to consider (row names of beta) (Default: NULL).

`samples`
: Subset of samples to consider (column names of beta) (Default: NULL).

`autosomal`
: If true, remove probes on sex chromosomes (Default: TRUE).

`winsorize.pct`
: Apply to methylation levels winsorized to the given level. Set to NA to avoid winsorizing (Default: NA).

`outlier.iqr.factor`
: Apply to methylation after setting, for each CpG site, values less than `Q1 - outlier.iqr.factor * IQR` or more than `Q3 + outlier.iqr.factor * IQR` to NA. Here IQR is the inter-quartile range of the methylation levels at the CpG site, i.e. Q3-Q1. Set to NA to skip this step (Default: NA).

Value

The n CpG site identifiers (rownames of x) with the greatest variance in x.

`meffil.normalization.parameters`
Specify parameters for testing normalization

Description

Specify parameters for testing normalization

Usage

```
meffil.normalization.parameters(
  norm.objects,
  variables = guess.batch.vars(norm.objects),
  control.pcs = 1:10,
  batch.pcs = 1:10,
  batch.threshold = 1e-50,
  colours = NULL
)
```

Arguments

`norm.objects`	Output from meffil.normalize.quantiles
`variables`	Default = guess.batch.vars(norm). Which variables in sample sheet to test
`control.pcs`	Default = 1:10. Number of control PCs to test against batch variables
`colours`	Colours to use for scatterplots.

Value

List of parameters

`meffil.normalization.parameters.from.betas`
Specify parameters for testing normalization

Description

Specify parameters for testing normalization

Usage

```
meffil.normalization.parameters.from.betas(
  batch.pcs = 1:10,
  batch.threshold = 1e-50,
  colours = NULL
)
```

Arguments

batch.pcs	Default = 1:10. Number of PCs to test against batch variables
batch.threshold	Default = 1e-50. Which pvalue threshold to show in table
colours	Colours to use for scatterplots.

Value

List of parameters

---

meffil.normalization.report

Generate report on normalization performance

Description

Generate HTML file that summarises the normalization.

Usage

```
meffil.normalization.report(
  normalization.summary,
  output.file = "normalization-report.md",
  author = "Analyst",
  study = "Illumina methylation data",
  ...
)
```

Arguments

normalization.summary	Output from meffil.normalization.summary.
output.file	Default = "meffil-normalization-report.html". If the file extension is not .htm, .html, .HTM or .HTML then output will be in markdown format.
author	Default = "Analyst". Author name to be specified on report.
study	Default = "Illumina methylation data". Study name to be specified on report.
...	Arguments to be passed to knitr::knit

meffil.normalization.report.from.betas

Generate report on normalization performance

Description

Generate HTML file that summarises the normalization.

Usage

```
meffil.normalization.report.from.betas(
  normalization.summary,
  output.file = "normalization-report.md",
  author = "Analyst",
  study = "Illumina methylation data",
  ...
)
```

Arguments

normalization.summary	Output from `meffil.normalization.summary.from.betas`.
output.file	Default = "meffil-normalization-report.html". If the file extension is not .htm, .html, .HTM or .HTML then output will be in markdown format.
author	Default = "Analyst". Author name to be specified on report.
study	Default = "Illumina methylation data". Study name to be specified on report.
...	Arguments to be passed to `knitr::knit`

meffil.normalization.summary

Perform tests to check normalization performance

Description

Creates scree plot of PCs of control probes, tests for association of control probe PCs with batch variables, tests for association of normalized probes with batch variables, creates PCA plots

Usage

```
meffil.normalization.summary(
  norm.objects,
  pcs,
  parameters = meffil.normalization.parameters(norm.objects),
  variables = NULL,
  verbose = TRUE
)
```

Arguments

norm.objects	Output from meffil.normalize.quantiles
pcs	Output from meffil.methylation.pcs() applied to the normalized methylation matrix corresponding to norm.objects.
parameters	Default = meffil.post.parameters(norm.objects). Report parameters.
variables	Default = NULL. Data frame of variables to compare to principal components (pcs). Must contain length(norm.objects) rows. Columns that are not factors are ignored.
verbose	Default = TRUE

Value

List of tables and graphs.

---

meffil.normalization.summary.from.betas

Perform tests to check normalization performance

Description

Perform tests to check normalization performance

Usage

```
meffil.normalization.summary.from.betas(
  pcs,
  parameters = meffil.normalization.parameters.from.betas(),
  samplesheet = samplesheet,
  variables = variables,
  verbose = TRUE
)
```

Arguments

pcs	Output from meffil.methylation.pcs() applied to the normalized methylation matrix
parameters	Default = meffil.normalization.parameters.from.betas(). Report parameters.
samplesheet	Default = NULL. Data frame of variables to compare to principal components (pcs). Must contain nrow(pcs) == nrow(samplesheet) rows. Columns that are not factors are ignored.
variables	Which variables in sample sheet to test
verbose	Default = TRUE

Value

List of tables and graphs.

`meffil.normalize.dataset` *Functional normalization*

Description

Apply functional normalization to a set of Infinium HumanMethylation450 BeadChip IDAT files.

Usage

```
meffil.normalize.dataset(
  samplesheet,
  number.quantiles = 500,
  detection.threshold = 0.01,
  bead.threshold = 3,
  sex.cutoff = -2,
  chip = NA,
  featureset = chip,
  cell.type.reference = NA,
  qc.parameters = meffil.qc.parameters(),
  qc.file = "meffil-qc-report.md",
  author = "Analyst",
  study = "IlluminaHuman450 data",
  number.pcs = 2,
  fixed.effects = NULL,
  random.effects = NULL,
  pseudo = 100,
  dup.fun = function(x) median(x, na.rm = T),
  just.beta = T,
  gds.filename = NULL,
  probe.range = 5000,
  autosomal = T,
  norm.parameters = NULL,
  norm.file = "meffil-normalization-report.md",
  verbose = FALSE
)
```

Arguments

`samplesheet`	Output from `meffil.read.samplesheet()` or `meffil.create.samplesheet()`.
	Arguments to `meffil.qc()`:
`cell.type.reference`	
	Argument to `meffil.qc.summary()`:
`qc.parameters`	(parameters)
	Arguments to `meffil.qc.report()`:
`qc.file`	(output.file)
`study`	Arguments to `meffil.normalize.quantiles()`:
`fixed.effects`	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).

random.effects	Names of columns in samplesheet that should be included as random effects (Default: NULL).
	Arguments to meffil.normalize.samples():
pseudo	Arguments to meffil.methylation.pcs().
dup.fun	Function to collapse duplicate probes (EPIC v2 has over 5000 duplicated probes). If NULL, then duplicates are not collapsed (Default: median).
gds.filename	If not NULL (default), then saves the output to a GDS (Genomic Data Structure). This is for cases where the output is too large to fit into main memory. The GDS option assumes that argument just.beta == TRUE.
probe.range	(Default: 5000).
autosomal	(Default: TRUE).
	Arguments to meffil.normalization.summary():
norm.parameters	(parameters)
norm.file	(output.file)
	Other:
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).
npcs	(Default: 1:10).

Details

Fortin JP, Labbe A, Lemire M, Zanke BW, Hudson TJ, Fertig EJ, Greenwood CM, Hansen KD. Functional normalization of 450k methylation array data improves replication in large cancer studies. Genome Biol. 2014 Dec 3;15(12):503. doi: 10.1186/s13059-014-0503-2. PMID: 25599564

Value

A list:

- qc.summary meffil.qc.summary() output.
- norm meffil.normalize.quantiles() output.
- beta Normalized beta matrix (methylation levels).
- norm.summary meffil.normalization.summary() output.

---

meffil.normalize.quantiles

Normalize microarray quantiles

Description

Normalize microarray quantiles using controls extracted (Infinium HumanMethylation450 BeadChip).

Usage

```
meffil.normalize.quantiles(
  qc.objects,
  number.pcs = 2,
  fixed.effects = NULL,
  random.effects = NULL,
  verbose = F
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
number.pcs	Number of control matrix principal components to adjust for (Default: 2).
fixed.effects	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).
random.effects	Names of columns in samplesheet that should be included as random effects (Default: NULL).
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

Same list as input with additional elements added for each sample including normalized quantiles needed for normalizing each sample.

---

meffil.normalize.sample

Normalize Infinium HumanMethylation450 BeadChip

Description

Normalize sample methylation data using normalized quantiles.

Usage

```
meffil.normalize.sample(norm.object, remove.poor.signal = F, verbose = F)
```

Arguments

norm.object	An element of meffil.normalize.quantiles().
remove.poor.signal	Set methylation values for poorly detected probes to missing (Default: FALSE). Poor signal was identified during QC by meffil.qc() as signal that failed to pass the detection p-value threshold (detection.threshold) or bead threshold (bead.threshold).
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

List containing normalized methylated and unmethylated signals.

meffil.normalize.samples *Normalize Infinium HumanMethylation450 BeadChips*

Description

Normalize a set of samples using their normalized quality control objects.

Usage

```
meffil.normalize.samples(
  norm.objects,
  pseudo = 100,
  just.beta = T,
  cpglist.remove = NULL,
  remove.poor.signal = F,
  dup.fun = function(x) median(x, na.rm = T),
  max.bytes = 2^30 - 1,
  gds.filename = NULL,
  verbose = F,
  ...
)
```

Arguments

norm.objects	The list or sublist of meffil.normalize.quantiles().
pseudo	Value to add to the denominator to make the methylation estimate more stable when calculating methylation levels (Default: 100).
just.beta	If TRUE, then return just the normalized methylation levels; otherwise, return the normalized methylated and unmethylated matrices (Default: TRUE).
cpglist.remove	Optional list of CpGs to exclude from final output
remove.poor.signal	Set methylation values for poorly detected probes to missing (Default: FALSE). Poor signal was identified during QC by meffil.qc() as signal that failed to pass the detection p-value threshold (detection.threshold) or bead threshold (bead.threshold).
dup.fun	Function to collapse duplicate probes (EPIC v2 has over 5000 duplicated probes). If NULL, then duplicates are not collapsed (Default: median).
gds.filename	If not NULL (default), then saves the output to a GDS (Genomic Data Structure). This is for cases where the output is too large to fit into main memory. The GDS option assumes that argument just.beta == TRUE.
verbose	If TRUE, then detailed status messages are printed during execution (Default: FALSE).
...	Arguments passed to mclapply().

Value

If `just.beta == TRUE`, the normalized matrix of methylation levels between between 0 and 1 equal to methylated signal/(methylated + unmethylated signal + pseudo). Otherwise, a list containing two matrices, the normalized methylated and unmethylated signals. If `gds.filename` is not `NULL`, then the output is saved to the GDS file rather than retained in memory and returned to the caller. The library 'gdsfmt' must be installed in this case.

`meffil.pcs` *Calculate control probe PCs*

Description

Calculate control probe PCs

Usage

```
meffil.pcs(qc.objects, fixed.effects = NULL, random.effects = NULL)
```

Arguments

`qc.objects`	A list of outputs from `meffil.create.qc.object()`.
`fixed.effects`	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).
`random.effects`	Names of columns in samplesheet that should be included as random effects (Default: NULL).

Value

PCA of control probes

`meffil.plot.beadnum.cpgs` *Manhattan plot of number of beads by probe - percentage of probes with beads < 3 for each sample*

Description

Manhattan plot of number of beads by probe - percentage of probes with beads < 3 for each sample

Usage

```
meffil.plot.beadnum.cpgs(qc.objects, threshold = 0.05)
```

Arguments

`qc.objects`	From `meffil.qc`
`threshold`	Cut off value for proportion of samples with poor detection p values. Default 0.05.

Value

Data frame of results plus plot

`meffil.plot.beadnum.samples`
Plot number of beads per sample

Description

Plot number of beads per sample

Usage

```
meffil.plot.beadnum.samples(qc.objects, threshold = 0.05, colour.code = NULL)
```

Arguments

`qc.objects`	From `meffil.qc`
`threshold`	Cut off value for proportion of CpGs with low bead numbers. Default 0.05
`colour.code`	Array of length n samples to colour code points. Defaults to NULL

Value

Data frame of results plus plot

`meffil.plot.cell.counts`
Cell count estimate quality plot

Description

Cell count estimate quality plot

Usage

```
meffil.plot.cell.counts(qc.objects)
```

Arguments

`qc.objects`	Output from `meffil.qc()`.
`reference`	Object describing methylation profiles of purified cell populations obtained from `meffil.add.cell.type.reference()`.

Value

Two ggplot2 boxplot objects:

- `betas` Contains one box per sample or reference cell type representing the distribution of methylation levels for the CpG sites used to estimate cell counts.
- `counts` Contains one box per reference cell type representing the distribution of cell count estimates across the samples.

meffil.plot.control.batch

Test for association of control matrix probes with known batch variables

Description

Performs association of each of n PCs calculated from the control matrix against each of m measured batch variables

Usage

```
meffil.plot.control.batch(
  norm.objects,
  npcs = 1:10,
  variables = guess.batch.vars(norm.objects),
  additional = NULL,
  batch.threshold = 1e-50,
  cols = NULL,
  verbose = TRUE
)
```

Arguments

norm.objects	From meffil.normalize.quantiles
pcs	Which PCs to plot. Default first 10
variables.	Default = guess.batch.vars(norm.objects). Array spacifying column names in samplesheet to test for association with control matrix PCs.
additional.	Default = NULL. Data frame containing variables to test for association with control matrix PCs. Must have nrow(additional) == length(norm.objects).
verbose=T	Print progress messages?

Value

Data frame of results plus plot

meffil.plot.control.scree

Plot scree plot of control matrix

Description

Plot scree plot of control matrix

Usage

```
meffil.plot.control.scree(norm.objects)
```

Arguments

norm.objects	From meffil.normalize.quantiles

Value

Data frame of results plus plot

---

meffil.plot.controlmeans

Plot the means of control probes for each sample and for each control probe type

---

Description

Plot the means of control probes for each sample and for each control probe type

Usage

```
meffil.plot.controlmeans(
  qc.objects,
  control.categories = NULL,
  colour.code = NULL,
  outlier.sd = 5
)
```

Arguments

qc.objects	From meffil.qc
control.categories	Which control probe categories to plot. Defaults to all available.
colour.code	Array of length n samples to colour code points. Defaults to NULL
outlier.sd	Cut off for declaring outliers. Default = 5

Value

Data frame of results plus plot

---

meffil.plot.detectionp.cpgs

Manhattan plot of detection pval per probe - percentage with pvalue < 0.01

Description

Manhattan plot of detection pval per probe - percentage with pvalue < 0.01

Usage

```
meffil.plot.detectionp.cpgs(qc.objects, threshold = 0.05)
```

Arguments

qc.objects	From meffil.qc
threshold	Cut off value for proportion of samples with poor detection p values. Default 0.05.

Value

Data frame of results plus plot

---

meffil.plot.detectionp.samples *Plot detection p values from idat files*

Description

Plot detection p values from idat files

Usage

```
meffil.plot.detectionp.samples(
  qc.objects,
  threshold = 0.05,
  colour.code = NULL
)
```

Arguments

qc.objects	From meffil.qc
threshold	Cut off value for proportion of CpGs with poor detection p values. Default 0.05
colour.code	Array of length n samples to colour code points. Defaults to NULL

Value

Data frame of results plus plot

---

meffil.plot.genotypes *Plot SNP beta and sample genotype concordances*

Description

Plot SNP beta and sample genotype concordances

Usage

```
meffil.plot.genotypes(
  qc.objects,
  genotypes = NULL,
  sample.threshold = 0.9,
  snp.threshold = 0.99
)
```

Arguments

qc.objects	Output from meffil.qc().
genotypes	Optional output from meffil.extract.genotypes(). Sample genotypes are matched to sample qc.objects using colnames(genotypes) and names(qc.objects).
sample.threshold	Concordance threshold below which the Illumina 450K and genetic profiles for a sample are deemed a mismatch (Default: 0.9).
snp.threshold	Concordance threshold below which the Illumina 450K and genetic profiles for a SNP are deemed a mismatch (Default: 0.99).

Value

A list consisting of:

- graphs A list of ggplot2 objects. The first snp.betas plots the beta distributions of each SNP probe in the microarray. The second and third plots are added only if the genotypes parameter is not NULL. The second plot shows the distribution of SNP concordances, and the third plot shows the distribution of sample concordances.
- tabs Contains two data frames if the genotypes parameter is not NULL. The first samples lists the concordances of each sample, the second snps lists the concordances of each SNP.

---

meffil.plot.meth.unmeth

Plot average methylated vs unmethylated levels for each individuals

---

Description

plot raw control probes and fit linear regression, remove samples that have sd(y - yhat) > mean*3

Usage

```
meffil.plot.meth.unmeth(qc.objects, outlier.sd = 3, colour.code = NULL)
```

Arguments

qc.objects	From meffil.qc
outlier.sd	Cut off for declaring outliers. Default = 3
colour.code	Array of length n samples to colour code points. Defaults to NULL

Value

Data frame of results plus plot

meffil.plot.pc.fit *Number of control matrix principal components*

Description

Fits probe intensities to principal components of the microarray control matrix and calculates the resulting mean squared residuals for different numbers of principal components.

Usage

```
meffil.plot.pc.fit(
  qc.objects,
  fixed.effects = NULL,
  random.effects = NULL,
  n.cross = 10,
  name = "autosomal.ii"
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
fixed.effects	Names of columns in samplesheet that should be included as fixed effects along with control matrix principal components (Default: NULL).
random.effects	Names of columns in samplesheet that should be included as random effects (Default: NULL).
number.pcs	Number of principal components to include in the design matrix (Default: all).

Value

A list containing a data frame with the mean squared residuals for different numbers of principal components and a plot of these residuals.

meffil.plot.probe.batch *Test normalized betas for association with known batch variables*

Description

Performs association of each of n PCs calculated from most variable CpG sites (after normalization) against each of m measured batch variables

Usage

```
meffil.plot.probe.batch(
  norm.objects,
  pcs,
  variables = guess.batch.vars(norm.objects),
  additional = NULL,
  batch.threshold = 1e-50,
  cols = NULL,
  verbose = T
)
```

Arguments

norm.objects	Output from meffil.normalize.quantiles().
pcs	Output from meffil.methylation.pcs() applied to the normalized methylation matrix corresponding to norm.objects.
variables	Default = guess.batch.vars(norm). Which variables in sample sheet to test
additional.	Default = NULL. Data frame containing variables to test for association with control matrix PCs. Must have nrow(additional) == length(norm.objects).
verbose=T	Print progress messages?

Value

List of table of results and graph

---

meffil.plot.probe.batch.from.betas

Test normalized betas for association with known batch variables

---

Description

Performs association of each of n PCs calculated from most variable CpG sites (after normalization) against each of m measured batch variables

Usage

```
meffil.plot.probe.batch.from.betas(
  samplesheet,
  variables,
  pcs,
  batch.threshold = batch.threshold,
  cols = NULL,
  verbose = T
)
```

Arguments

variables	Which variables in sample sheet to test
pcs	Output from meffil.methylation.pcs() applied to the normalized methylation matrix
samplesheet.	Data frame containing variables to test for association with PCs. Must have nrow(pcs) == nrow(samplesheet).
verbose=T	Print progress messages?

Value

List of table of results and graph

---

meffil.plot.sex *Plot predicted sex*

Description

Plot predicted sex

Usage

```
meffil.plot.sex(qc.objects, outlier.sd = 3)
```

Arguments

qc.objects	From meffil.qc

Value

Data frame of results plus plot

---

meffil.probe.info *Obtain a list of probes for a given feature set (chip).*

Description

Obtain a list of probes for a given feature set (chip).

Usage

```
meffil.probe.info(chip = "450k", featureset = chip)
```

Arguments

chip	Name returned by meffil.list.chips() (Default: "450k").
featureset	Name returned by meffil.list.featuresets() (Default: chip).

Value

A data frame with one row per probe. The full set of probes for a chip is returned if chip == featureset; otherwise, the probes are restricted to those corresponding to features in the feature set.

meffil.qc *Perform QC on HumanMethylation450 idat files*

Description

Read in control matrices for each sample. Perform background correction and R/G dye bias correction. Predict sex

Usage

```
meffil.qc(
  samplesheet,
  number.quantiles = 500,
  dye.intensity = 5000,
  detection.threshold = 0.01,
  bead.threshold = 3,
  sex.cutoff = -2,
  chip = NA,
  featureset = chip,
  cell.type.reference = NA,
  max.bytes = 2^30 - 1,
  verbose = F,
  ...
)
```

Arguments

samplesheet	Data frame containing IDAT file and sample info (see meffil.read.samplesheet pr meffil.create.samplesheet).
number.quantiles	Number of quantiles to compute for probe subset (Default: 500).
dye.intensity	Reference intensity for scaling each color channel (Default: 5000).
detection.threshold	Default value = 0.01. All probes above this detection threshold detected.
bead.threshold	Default value = 3. All probes with less than this number of beads detected.
sex.cutoff	Sex prediction cutoff. Default value = -2.
chip	Name returned by meffil.list.chips() (Default: NA).
featureset	Name returned by meffil.list.featuresets() (Default: chip).
cell.type.reference	Character string name of the cell type reference to use for estimating cell counts. Estimates are not generated if set to NA (default). See meffil.list.cell.type.references() for a list of available references. New references can be created using meffil.add.cell.type.refer
verbose	If TRUE, then status messages are printed during execution (Default: FALSE).

Value

List containing control probe information, probe summaries and quantiles.

`meffil.qc.parameters` *Specify parameters for QC*

Description

Specify parameters for QC

Usage

```
meffil.qc.parameters(
  colour.code = NULL,
  control.categories = NULL,
  sex.outlier.sd = 3,
  meth.unmeth.outlier.sd = 3,
  control.means.outlier.sd = 5,
  detectionp.samples.threshold = 0.2,
  beadnum.samples.threshold = 0.2,
  detectionp.cpgs.threshold = 0.2,
  beadnum.cpgs.threshold = 0.2,
  snp.concordance.threshold = 0.9,
  sample.genotype.concordance.threshold = 0.9
)
```

Arguments

`colour.code` Default value = NULL

`control.categories`
Default value = control.probe.categories()

`sex.outlier.sd` Sets the standard deviation multiple at which sex outliers are identified. Default value = 3.

`meth.unmeth.outlier.sd`
Sets the standard deviation multiple at which methylated/unmethylated signal outliers are identified. Default value = 3.

`control.means.outlier.sd`
Sets the standard deviation multiple at which control probe signals are identified as outliers. Default value = 5

`detectionp.samples.threshold`
Maximum threshold on the fraction of undetected probes (probe detection is defined by setting the maximum probe detection p-value threshold parameter `detection.threshold` of `meffil.qc()` or `meffil.normalize.dataset()`). Samples with probe fractions above this will be excluded from the final dataset. Default value = 0.2

`beadnum.samples.threshold`
Maximum threshold on the fraction of probes with too few detected beads (minimum number of detected beads is defined by setting the `beads.threshold` parameter of `meffil.qc()` or `meffil.normalize.dataset()`). Samples with probe fractions above this will be excluded from the final dataset. Default value = 0.2

`detectionp.cpgs.threshold`
: Same as `detectionp.cpgs.threshold` but used to identify poor quality probes in terms of the fraction of samples in which the probe is undetected. Default value = 0.2

`beadnum.cpgs.threshold`
: Same as `beadnum.samples.threshold` but used to identify poor quality probes in terms of the fraction of samples in which the probe has too few detected beads. Default value = 0.2

`snp.concordance.threshold`
: Minimum required concordance between supplied genotypes and genotypes estimated from a SNP probe. Default value = 0.99

`sample.genotype.concordance.threshold`
: Minimum required concordance between supplied genotypes and genotypes estimated from SNP probes for a given individual. Default value = 0.9

Value

List of parameter values

`meffil.qc.report` *Generate QC report*

Description

Generate HTML file that summarises the QC.

Usage

```
meffil.qc.report(
  qc.summary,
  output.file = "qc-report.html",
  author = "Analyst",
  study = "Illumina methylation data",
  ...
)
```

Arguments

`qc.summary`	Output from `meffil.qc.summary`.
`output.file`	Default = "meffil-qc-report.html". If the file extension is not .htm, .html, .HTM or .HTML then output will be in markdown format.
`author`	Default = "Analyst". Author name to be specified on report.
`study`	Default = "Illumina methylation data". Study name to be specified on report.
`...`	Arguments to be passed to `knitr::knit`

`meffil.qc.summary` *Perform QC analysis on idat files*

Description

Performs a number of QC analyses including checking for sex differences, methylated vs unmethylated levels, deviation from control probe means, detection p-values and bead numbers per sample and probe.

Usage

```
meffil.qc.summary(
  qc.objects,
  genotypes = NULL,
  parameters = meffil.qc.parameters(),
  verbose = TRUE
)
```

Arguments

`qc.objects`	From `meffil.qc`
`genotypes`	Optional output from `meffil.extract.genotypes()`. Sample genotypes are matched to sample qc.objects using `colnames(genotypes)` and `names(qc.objects)`.
`parameters`	Default = meffil.qc.parameters(). List of parameter values. See `meffil.qc.parameters`

Details

Also returns list of sample IDs and CPGs that are low quality.

Value

List

`meffil.read.samplesheet`

Function to read Illumina "Sample Sheet" adapted from `read.450k.sheet` *in* `R/minfi`

Description

Reading an Illumina methylation sample sheet, containing pheno-data information for the samples in an experiment.

Usage

```
meffil.read.samplesheet(
  base,
  pattern = "csv$",
  ignore.case = TRUE,
  recursive = TRUE,
  verbose = TRUE
)
```

Arguments

base	The base directory from which the search is started.
pattern	= "csv$" What pattern is used to identify a sample sheet file, see list.files
ignore.case	= TRUE Should the file search be case sensitive?
recursive	= TRUE Should the file search be recursive, see list.files?
verbose	= TRUE Should the function be verbose?
basenames	Output from meffil.basenames

Details

This function search the directory base (possibly including subdirectories depending on the argument recursive for "sample sheet" files (see below). These files are identified solely on the base of their filename given by the arguments pattern and ignore.case (note the use of a dollarsign to mean end of file name).#'

In case multiple sheet files are found, they are all read and the return object will contain the concatenation of the files.

A sample sheet file is essentially a CSV (comma-separated) file containing one line per sample, with a number of columns describing pheno-data or other important information about the sample. The file may contain a header, in which case it is assumed that all lines up to and including a line starting with \[Data\] should be dropped. This is modelled after a sample sheet file Illumina provides. It is also very similar to the targets file made used by the popular limma package (see the extensive package vignette).#'

An attempt at guessing the file path to the IDAT files represented in the sheet is made. This should be doublechecked and might need to manually changed.

Value

A data.frame containing the columns of all the sample sheets. As described in details, a column named Sentrix_Position is renamed to Array and Sentrix_ID is renamed to Slide. In addition the data.frame will contain a column named Basename.

---

meffil.remove.samples *Remove samples from QC objects*

Description

Remove samples from QC objects

Usage

```
meffil.remove.samples(qc.objects, sample.ids)
```

Arguments

qc.objects	Output from meffil.qc
sample.ids	Array of sample.name IDs to be removed

Value

qc.objects with samples removed

meffil.save.detection.pvalues *Save detection p-value matrix to GDS file*

Description

Save detection p-value matrix to GDS file

Usage

```
meffil.save.detection.pvalues(
  qc.objects,
  gds.filename = NULL,
  max.bytes = 2^30 - 1,
  verbose = F,
  ...
)
```

Arguments

qc.objects	A list of outputs from meffil.create.qc.object().
gds.filename	If not NULL (default), then saves the output to a GDS (Genomic Data Structure). This is for cases where the output is too large to fit into main memory.
verbose	If TRUE, then detailed status messages are printed during execution (Default: FALSE).
...	Arguments passed to mclapply().

Value

Matrix of probe detection p-values. If gds.filename is not NULL, then the output is saved to the GDS file rather than retained in memory and returned. The library 'gdsfmt' must be installed in this case.

meffil.snp.betas *Matrix of SNP 'beta' values*

Description

Matrix of SNP 'beta' values

Usage

```
meffil.snp.betas(qc.objects)
```

Arguments

qc.objects	List of objects obtained from meffil.qc() or meffil.create.qc.object().

`meffil.snp.concordance` *Concordance between genotypes and SNP betas*

Description

genotypes <- meffil.extract.genotypes(raw.filenames) snp.betas <- meffil.snp.betas(qc.objects) meffil.snp.concordance(snp.betas, genotypesrownames(snp.betas),colnames(snp.betas))

Usage

```
meffil.snp.concordance(
  snp.betas,
  genotypes,
  snp.threshold = 0.99,
  sample.threshold = 0.9
)
```

Value

Returns a list of two vectors: - one providing concordances between genotypes and SNP betas for matched samples, - a second providing concordances between genotypes and SNP betas for matched SNPs. as well as the genotype matrix derived from 'snp.betas'.

`meffil.snp.names` *Obtain the list of identifiers for the SNPs on the microarray.*

Description

Obtain the list of identifiers for the SNPs on the microarray.

Usage

```
meffil.snp.names(featureset = "450k")
```

Arguments

`featureset`	Name from `meffil.list.featuresets()` (Default: "450k").

`meffil.summarize.relationship` *Describe the relationship between two variables.*

Description

Describe the relationship between two variables.

Usage

```
meffil.summarize.relationship(vars)
```

Arguments

`vars`	A data frame with at least two columns. The first two columns will be compared.

Value

A list whose elements depends on the types of the two variables. In each case, the list contains the following elements:

var1 Name of the first variable, i.e. colnames(vars)[1].

var2 Name of the second variable.

r Spearman's correlation between the variables. This may be meaningless if one variable is an unordered factor.

r.p P-value corresponding to the correlation between the variables.

output The contents of the list formatted to be printed as markdown text.

plot A plot (ggplot2) visualizing the relationship.

If both variables are factors, then the list will include a frequency table (`freq`) and a corresponding matrix of p-values (`p.values`) obtained using Fisher's test to test for enrichment in each cell of the frequency table.

If one variable is a factor and the other numeric, then list will include the F-statistic (`F.stat`) and p-value (`p.value`) from one-way analysis of variance. There will also be a data frame (`cases`) with each row providing statistics from a t-test comparing the numeric variable within and without each level of the factor variable.

If both variables are numeric, then the list will include the F-statistic (`F.stat`) and p-value (`p.value`) from the linear model fitting the variables.

Index