Package 'BPRMeth'

April 14, 2017

Type Package

Title Model higher-order methylation profiles

Version 1.0.0

Author Chantriolnt-Andreas Kapourani [aut, cre]

Maintainer Chantriolnt-Andreas Kapourani

<kapouranis.andreas@gmail.com>

Description BPRMeth package uses the Binomial Probit Regression likelihood to model methylation profiles and extract higher order features. These features quantitate precisely notions of shape of a methylation profile. Using these higher order features across promoter-proximal regions, we construct a powerful predictor of gene expression. Also, these features are used to cluster proximal-promoter regions using the EM algorithm.

Depends R (>= 3.3.0), GenomicRanges

License GPL-3

LazyData true

RoxygenNote 5.0.1

Imports assertthat, methods, MASS, doParallel, parallel, e1071, earth, foreach, randomForest, stats, IRanges, S4Vectors, data.table, graphics

Suggests testthat, knitr, rmarkdown, BiocStyle

VignetteBuilder knitr

biocViews DNAMethylation, GeneExpression, GeneRegulation, Epigenetics, Genetics, Clustering, FeatureExtraction, Regression, RNASeq, Bayesian, KEGG, Sequencing, Coverage

NeedsCompilation no

R topics documented:

`boxplot_cluster_gex` *Boxplot of clustered expression levels*

Description

`boxplot_cluster_gex` creates a boxplot of clustered gene expression levels which depend on the clustered methylation profiles. Each colour denotes a different cluster.

Usage

```
boxplot_cluster_gex(bpr_cluster_obj, gex, main_lab = "Gene expression levels")
```

Arguments

`bpr_cluster_obj`	The output of the `bpr_cluster_wrap` function.
`gex`	The vector of gene expression data for each promoter region.
`main_lab`	The title of the plot

Value

The figure to be plotted in the device.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

`plot_cluster_prof`, `plot_scatter_gex`, `plot_fitted_profiles`

Examples

```
# Cluster methylation profiles using 4 RBFs
obs <- meth_data
basis <- create_rbf_object(M = 4)
res   <- bpr_cluster_wrap(x = obs, K = 3, em_max_iter = 5, opt_itnmax = 4,
                          init_opt_itnmax = 5, is_parallel = FALSE)

# Create the plot
boxplot_cluster_gex(bpr_cluster_obj = res, gex = gex_data)
```

BPRMeth *BPRMeth: Extracting higher order methylation features*

Description

Higher order methylation features for clustering and prediction in epigenomic studies

Usage

```
.datatable.aware
```

Format

An object of class `logical` of length 1.

Value

BPRMeth main package documentation.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

---

bpr_cluster_wrap *Cluster methylation profiles*

Description

`bpr_cluster_wrap` is a wrapper function that clusters methylation profiles using the EM algorithm. Initially, it performs parameter checking, and initializes main parameters, such as mixing proportions, basis function coefficients, then the EM algorithm is applied and finally model selection metrics are calculated, such as BIC and AIC.

Usage

```
bpr_cluster_wrap(x, K = 3, pi_k = NULL, w = NULL, basis = NULL,
  em_max_iter = 100, epsilon_conv = 1e-04, opt_method = "CG",
  opt_itnmax = 100, init_opt_itnmax = 100, is_parallel = TRUE,
  no_cores = NULL, is_verbose = FALSE)
```

Arguments

x	The binomial distributed observations, which has to be a list of elements of length N, where each element is an L x 3 matrix of observations, where 1st column contains the locations. The 2nd and 3rd columns contain the total trials and number of successes at the corresponding locations, repsectively. See process_haib_caltech_wrap on a possible way to get this data structure.
K	Integer denoting the number of clusters K.
pi_k	Vector of length K, denoting the mixing proportions.
w	A MxK matrix, where each column consists of the basis function coefficients for each corresponding cluster.
basis	A 'basis' object. E.g. see create_rbf_object.
em_max_iter	Integer denoting the maximum number of EM iterations.
epsilon_conv	Numeric denoting the convergence parameter for EM.
opt_method	The optimization method to be used. See optim for possible methods. Default is "CG".
opt_itnmax	Optional argument giving the maximum number of iterations for the corresponding method. See optim for details.
init_opt_itnmax	Optimization iterations for obtaining the initial EM parameter values.
is_parallel	Logical, indicating if code should be run in parallel.
no_cores	Number of cores to be used, default is max_no_cores - 2.
is_verbose	Logical, print results during EM iterations.

Value

A 'bpr_cluster' object which, in addition to the input parameters, consists of the following variables:

- pi_k: Fitted mixing proportions.
- w: A MxK matrix with the fitted coefficients of the basis functions for each cluster k.
- NLL: The Negative Log Likelihood after the EM algorithm has finished.
- post_prob: Posterior probabilities of each promoter region belonging to each cluster.
- labels: Hard clustering assignments of each observation/promoter region.
- BIC: Bayesian Information Criterion metric.
- AIC: Akaike Information Criterion metric.
- ICL: Integrated Complete Likelihood criterion metric.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

Examples

```
ex_data <- meth_data
data_clust <- bpr_cluster_wrap(x = ex_data, em_max_iter = 3, opt_itnmax = 5,
                               init_opt_itnmax = 10, is_parallel = FALSE)
```

bpr_optimize *Optimize BPR negative log likelihood function*

Description

The function bpr_optimize minimizes the negative log likelihood of the BPR function. Since it cannot be evaluated analytically, an optimization procedure is used. The optim packages is used for performing optimization.

Usage

```
bpr_optim(x, ...)

## S3 method for class 'list'
bpr_optim(x, w = NULL, basis = NULL, fit_feature = "RMSE",
  cpg_dens_feat = TRUE, opt_method = "CG", opt_itnmax = 100,
  is_parallel = TRUE, no_cores = NULL, ...)

## S3 method for class 'matrix'
bpr_optim(x, w = NULL, basis = NULL,
  fit_feature = "RMSE", cpg_dens_feat = TRUE, opt_method = "CG",
  opt_itnmax = 100, ...)
```

Arguments

x	The input object, either a matrix or a list.
...	Additional parameters.
w	A vector of parameters (i.e. coefficients of the basis functions)
basis	A 'basis' object. E.g. see create_rbf_object.
fit_feature	Return additional feature on how well the profile fits the methylation data. Either NULL for ignoring this feature or one of the following: 1) "RMSE" for returning the fit of the profile using the RMSE as measure of error or 2) "NLL" for returning the fit of the profile using the Negative Log Likelihood as measure of error.
cpg_dens_feat	Logical, whether to return an additional feature for the CpG density across the promoter region.
opt_method	The optimization method to be used. See optim for possible methods. Default is "CG".
opt_itnmax	Optional argument giving the maximum number of iterations for the corresponding method. See optim for details.
is_parallel	Logical, indicating if code should be run in parallel.
no_cores	Number of cores to be used, default is max_no_cores - 2.

Value

Depending on the input object x:

- If x is a list: An object containing the following elements:

  - W_opt: An Nx(M+1) matrix with the optimized parameter values. Each row of the matrix corresponds to each element of the list x. The columns are of the same length as the parameter vector w (i.e. number of basis functions).
  - Mus: An N x M matrix with the RBF centers if basis object is create_rbf_object, otherwise NULL.
  - basis: The basis object.
  - w: The initial values of the parameters w.
- If x is a matrix: An object containing the following elements:
  - w_opt: Optimized values for the coefficient vector w. The length of the result is the same as the length of the vector w.
  - basis: The basis object.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

create_basis, eval_functions

Examples

```
# Example of optimizing parameters for synthetic data using default values
data <- meth_data
out_opt <- bpr_optim(x = data, is_parallel = FALSE, opt_itnmax = 10)

#-------------------------------------

# Example of optimizing parameters for synthetic data using 3 RBFs
ex_data <- meth_data
basis <- create_rbf_object(M=3)
out_opt <- bpr_optim(x = ex_data, is_parallel = FALSE, basis = basis,
                     opt_itnmax = 10)

#-------------------------------------

# Example of of specific promoter region using 2 RBFs
basis <- create_rbf_object(M=2)
w <- c(0.1, 0.1, 0.1)
data <- meth_data[[1]]
out_opt <- bpr_optim(x = data, w = w, basis = basis, fit_feature = "NLL",
                     opt_itnmax = 10)
```

---

bpr_predict_wrap *Predict gene expression from methylation profiles*

Description

bpr_predict_wrap is a function that wraps all the necessary subroutines for performing prediction on gene expression levels. Initially, it optimizes the parameters of the basis functions so as to learn the methylation profiles. Then, uses the learned parameters / coefficients of the basis functions as input features for performing regression in order to predict the corresponding gene expression levels.

Usage

```
bpr_predict_wrap(formula = NULL, x, y, model_name = "svm", w = NULL,
  basis = NULL, train_ind = NULL, train_perc = 0.7,
  fit_feature = "RMSE", cpg_dens_feat = TRUE, opt_method = "CG",
  opt_itnmax = 100, is_parallel = TRUE, no_cores = NULL,
  is_summary = TRUE)
```

Arguments

formula	An object of class formula, e.g. see lm function. If NULL, the simple linear regression model is used.
x	The binomial distributed observations, which has to be a list of elements of length N, where each element is an L x 3 matrix of observations, where 1st column contains the locations. The 2nd and 3rd columns contain the total trials and number of successes at the corresponding locations, repsectively. See process_haib_caltech_wrap on a possible way to get this data structure.
y	Corresponding gene expression data for each element of the list x.
model_name	A string denoting the regression model. Currently, available models are: "svm", "randomForest", "rlm", "mars" and "lm".
w	Optional vector of initial parameter / coefficient values.
basis	Optional basis function object, default is an 'rbf' object, see create_rbf_object.
train_ind	Optional vector containing the indices for the train set.
train_perc	Optional parameter for defining the percentage of the dataset to be used for training set, the remaining will be the test set.
fit_feature	Return additional feature on how well the profile fits the methylation data. Either NULL for ignoring this feature or one of the following: 1) "RMSE" for returning the fit of the profile using the RMSE as measure of error or 2) "NLL" for returning the fit of the profile using the Negative Log Likelihood as measure of error.
cpg_dens_feat	Logical, whether to return an additional feature for the CpG density across the promoter region.
opt_method	The optimization method to be used. See optim for possible methods. Default is "CG".
opt_itnmax	Optional argument giving the maximum number of iterations for the corresponding method. See optim for details.
is_parallel	Logical, indicating if code should be run in parallel.
no_cores	Number of cores to be used, default is max_no_cores - 2.
is_summary	Logical, print the summary statistics.

Value

A 'bpr_predict' object which, in addition to the input parameters, consists of the following variables:

- W_opt: An Nx(M+1) matrix with the optimized parameter values. Each row of the matrix corresponds to each element of the list x. The columns are of the same length as the parameter vector w (i.e. number of basis functions).
- Mus: An N x M matrix with the RBF centers if basis object is create_rbf_object, otherwise NULL.

- train: The training data.
- test: The test data.
- gex_model: The fitted regression model.
- train_pred The predicted values for the training data.
- test_pred The predicted values for the test data.
- train_errors: The training error metrics.
- test_errors: The test error metrics.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

bpr_optimize, create_basis, eval_functions, train_model_gex, predict_model_gex

Examples

```
obs <- meth_data
y   <- gex_data
basis <- create_rbf_object(M = 5)
out   <- bpr_predict_wrap(x = obs, y = y, basis = basis,
                          is_parallel = FALSE, opt_itnmax = 10)
```

---

create_basis *Create basis objects*

Description

These functions create different basis objects. These objects can be used as input to complex functions in order to perform computations depending on the class of the basis function.

Usage

```
create_rbf_object(M = 2, gamma = NULL, mus = NULL, eq_spaced_mus = TRUE,
  whole_region = TRUE)

create_polynomial_object(M = 1)
```

Arguments

M	The number of the basis functions.
gamma	Inverse width of radial basis function.
mus	Optional centers of the RBF.
eq_spaced_mus	Logical, if TRUE, equally spaced centers are created, otherwise centers are created using kmeans algorithm.
whole_region	Logical, indicating if the centers will be evaluated equally spaced on the whole region, or between the min and max of the observation values.

Value

A basis object of class 'rbf' or 'polynomial'.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

eval_functions, bpr_optimize

Examples

```
(obj <- create_rbf_object(M = 2))

#---------------------------------

(obj <- create_polynomial_object(M = 2))
```

---

create_methyl_region *Create methylation regions for each gene promoter.*

Description

create_methyl_region creates methylation regions using BS-Seq and annotated gene promoter regions. BS-Seq data give information for the methylation of CpGs individually, and annotated data are used to locate the TSS of each gene and its promoter region.

Usage

```
create_methyl_region(bs_data, prom_region, cpg_density = 10,
  sd_thresh = 0.1, ignore_strand = TRUE, fmin = -1, fmax = 1)
```

Arguments

bs_data	GRanges object containing the BS-Seq data. The GRanges object should also have two additional metadata columns named total_reads and meth_reads. A GRanges object used in this function can be the output of read_bs_encode_haib or its wrapper function preprocess_bs_seq.
prom_region	GRanges object containing promoter regions, i.e. N bp upstream and M bp downstream of TSS location. The GRanges object should also have one additional metadata column named tss. A GRanges object used in this function can be the output of create_prom_region.
cpg_density	Optional integer defining the minimum number of CpGs that have to be in a methylated region. Regions with less than n CpGs are discarded.
sd_thresh	Optional numeric defining the minimum standard deviation of the methylation change in a region. This is used to filter regions with no methylation change.
ignore_strand	Logical, whether or not to ignore strand information.
fmin	Optional minimum range value for region location scaling. Under this version, this parameter should be left to its default value.

fmax	Optional maximum range value for region location scaling. Under this version, this parameter should be left to its default value.

Value

A methyl_region object containing the following information:

- meth_data: A list containing methylation data, where each entry in the list is an L_iX3 dimensional matrix, where L_i denotes the number of CpGs found in region i. The columns contain the following information:
  1. 1st column: Contains the locations of CpGs relative to TSS. Note that the actual locations are scaled to the (fmin, fmax) region.
  2. 2nd column: Contains the total reads of each CpG in the corresponding location.
  3. 3rd column: Contains the methylated reads each CpG in the corresponding location.
- prom_ind: A vector storing the corresponding promoter indices, so as to map each methylation region with its corresponding gene promoter.

The lengths of meth_data and prom_ind should be the same.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

preprocess_bs_seq, create_prom_region

Examples

```
# Obtain the path to the BS file and then read it
bs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
bs_data <- read_bs_encode_haib(bs_file)

# Create promoter regions
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
annot_data <- read_rna_encode_caltech(rnaseq_file)
prom_region <- create_prom_region(annot_data)

# Finally, create methylation regions
meth_region <- create_methyl_region(bs_data, prom_region)
```

---

create_prom_region *Create promoter regions from gene annotation data.*

Description

create_prom_region creates promoter region from gene annotation data. Using the TSS of gene annotation data as ground truth labels we create promoter regions N bp upstream and M bp downstream of TSS.

Usage

```
create_prom_region(annot_data, chrom_size = NULL, upstream = -7000,
  downstream = 7000)
```

Arguments

annot_data	A GRanges object containing the gene annotation data. This for example can be RNA-Seq data output from read_rna_encode_caltech.
chrom_size	Optional data.table containing chromosome sizes, e.g. using the read_chrom_size function.
upstream	Integer defining the length of bp upstream of TSS.
downstream	Integer defining the length of bp downstream of TSS.

Value

A GRanges object containing the promoter regions data.

The GRanges object contains one additional metadata column:

- tss: TSS of each gene promoter.

This column can be accessed as follows: granges_object$tss

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

create_methyl_region, read_chrom_size, read_rna_encode_caltech

Examples

```
# Obtain the path to the file and then read it
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
annot_data <- read_rna_encode_caltech(rnaseq_file)
prom_region <- create_prom_region(annot_data)

# Extract the TSS
tss <- prom_region$tss
```

---

eval_functions *Evaluate basis functions*

Description

Method for evaluating an M basis function model with observation data obs and coefficients w.

Usage

```
eval_probit_function(x, ...)

eval_function(x, ...)

## S3 method for class 'rbf'
eval_function(x, obs, w, ...)

## S3 method for class 'polynomial'
eval_function(x, obs, w, ...)
```

Arguments

x	The basis function object.
...	Optional additional parameters
obs	Observation data.
w	Vector of length M, containing the coefficients of an M^{th}-order basis function.

Value

The evaluated function values.

NOTE that the `eval_probit_function` computes the probit transformed basis function values.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

create_basis

Examples

```
# Evaluate the probit transformed basis function values
x <- create_rbf_object(M=2)
obs <- c(1,2,3)
w <- c(0.1, 0.3, -0.6)
out <- eval_probit_function(x, obs, w)

# -------------------------

# Evaluate the RBF basis function values
x <- create_rbf_object(M=2, mus = c(2,2.5))
obs <- c(1,2,3)
w <- c(0.1, 0.3, -0.6)
out <- eval_function(x, obs, w)

# -------------------------

# Evaluate the Polynomial basis function values
x <- create_polynomial_object(M=2)
obs <- c(1,2,3)
w <- c(0.1, 0.3, -0.6)
out <- eval_function(x, obs, w)
```

gex_data *Synthetic data for mpgex package*

Description

Corresponding gene expression data for the 'meth_data'

Usage

```
gex_data
```

Format

A vector of length 600

Value

Synthetic gene expression data

meth_data *Synthetic data for BPRMeth package*

Description

A synthetic dataset containinig 600 entries.

Usage

```
meth_data
```

Format

A list with 600 elements, where each element element is an L x 3 matrix of observations, where:

1st column locations of obseravtions

2nd column total trials at corresponding locations

3rd column number of successes at corresponding locations

Value

Synthetic methylation data

plot_cluster_prof *Plot of clustered methylation profiles*

Description

plot_cluster_prof creates a plot of cluster methylation profiles, where each colour denotes a different cluster.

Usage

```
plot_cluster_prof(bpr_cluster_obj,
  main_lab = "Clustered methylation profiles")
```

Arguments

bpr_cluster_obj	The output of the bpr_cluster_wrap function.
main_lab	The title of the plot

Value

The figure to be plotted in the device.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

plot_scatter_gex, plot_fitted_profiles, boxplot_cluster_gex

Examples

```
# Cluster methylation profiles using 4 RBFs
obs <- meth_data
basis <- create_rbf_object(M = 4)
res   <- bpr_cluster_wrap(x = obs, K = 3, em_max_iter = 5, opt_itnmax = 4,
                          init_opt_itnmax = 5, is_parallel = FALSE)

# Create the plot
plot_cluster_prof(bpr_cluster_obj = res)
```

`plot_fitted_profiles` *Plot the fit of methylation profiles across a region*

Description

`plot_fitted_profiles` is a simple function for plotting the methylation data across a give region, together with the fit of the methylation profiles.

Usage

```
plot_fitted_profiles(region, X, fit_prof, fit_mean = NULL,
  title = "Gene promoter", ...)
```

Arguments

`region`	Promoter region number
`X`	Methylation data observations
`fit_prof`	Fitted profile
`fit_mean`	Fitted mean function
`title`	Title of the plot
`...`	Additional parameters

Value

The figure to be plotted in the device.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

`plot_cluster_prof`, `plot_scatter_gex`, `boxplot_cluster_gex`

Examples

```
# Fit methylation profiles using 8 RBFs
obs <- meth_data
y   <- gex_data
basis <- create_rbf_object(M = 8)
out   <- bpr_predict_wrap(x = obs, y = y, basis = basis,
                          is_parallel = FALSE, opt_itnmax = 10)

# Create the plot
plot_fitted_profiles(region = 16, X = meth_data, fit_prof = out)
```

plot_scatter_gex *Scatter plot of predicted vs measured gene expression levels*

Description

plot_scatter_gex creates a scatter plot of predicted gene expression values on the x-axis versus the measured gene expression values on the y-axis.

Usage

```
plot_scatter_gex(bpr_predict_obj, main_lab = "Methylation Profile",
  is_margins = TRUE)
```

Arguments

bpr_predict_obj	The output of the bpr_predict_wrap function.
main_lab	The title of the plot
is_margins	Use specific margins or not.

Value

The figure to be plotted in the device.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

plot_cluster_prof, plot_fitted_profiles, boxplot_cluster_gex

Examples

```
# Fit methylation profiles using 8 RBFs
obs <- meth_data
y   <- gex_data
basis <- create_rbf_object(M = 8)
res   <- bpr_predict_wrap(x = obs, y = y, basis = basis,
                          is_parallel = FALSE, opt_itnmax = 10)

# Create the scatter plot
plot_scatter_gex(bpr_predict_obj = res)
```

pool_bs_seq_rep *Read and pool replicates from BS-Seq data*

Description

pool_bs_seq_rep reads and pools replicate methylation data from BS-Seq experiments that are either in Encode RRBS or Bismark Cov format. Read the Important section below on when to use this function.

Usage

```
pool_bs_seq_rep(files, file_format = "encode_rrbs", chr_discarded = NULL)
```

Arguments

files	A vector of filenames containing replicate experiments. This can also be just a single replicate.
file_format	A string denoting the file format that the BS-Seq data are stored. Current version allows "encode_rrbs" or "bismark_cov" formats.
chr_discarded	A vector with chromosome names to be discarded.

Value

A GRanges object. The GRanges object contains two additional metadata columns:

- total_reads: total reads mapped to each genomic location.
- meth_reads: methylated reads mapped to each genomic location.

These columns can be accessed as follows: granges_object$total_reads

Important

Unless you want to create a different workflow when processing the BS-Seq data, you should NOT call this function, since this is a helper function. Instead you should call the preprocess_bs_seq function.

Information about the file formats can be found in the following links:

Encode RRBS format: http://rohsdb.cmb.usc.edu/GBshape/cgi-bin/hgTables?db=hg19&hgta_group=regulation&hgta_track=wgEncodeHaibMethylRrbs&hgta_table=wgEncodeHaibMethylRrbsBcbreas
hgta_doSchema=describe+table+schema

Bismark Cov format: http://rnbeads.mpi-inf.mpg.de/data/RnBeads.pdf

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_bs_bismark_cov, read_bs_encode_haib, preprocess_bs_seq

Examples

```
# Obtain the path to the file
bs_file1 <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
bs_file2 <- system.file("extdata", "rrbs.bed", package = "BPRMeth")

# Concatenate the files
bs_files <-  c(bs_file1, bs_file2)
# Pool the replicates
pooled_data <- pool_bs_seq_rep(bs_files)
```

---

predict_model_gex *Predict gene expression model from methylation profiles*

Description

predict_model_gex makes predictions of gene expression levels using a model trained on higher order methylation features extracted from specific genomic regions.

Usage

```
predict_model_gex(model, test, is_summary = TRUE)
```

Arguments

model	The fitted regression model, i.e. the output of train_model_gex.
test	The testing data.
is_summary	Logical, print the summary statistics.

Value

A list containing the following elements:

- test_pred: The predicted values for the test data.
- test_errors: The test error metrics.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

train_model_gex

Examples

```
# Create synthetic data
train_data <- data.frame(x = rnorm(20), y=rnorm(20, 1, 4))
test_data <- data.frame(x = rnorm(20), y=rnorm(20, 1, 3))

# Train the model
train_model <- train_model_gex(formula = y~., train = train_data)

# Make predictions
res <- predict_model_gex(model = train_model$gex_model, test = test_data)
```

---

preprocess_bs_seq *Pre-process BS-Seq data in any given format*

Description

preprocess_bs_seq is a general function for reading and preprocessing BS-Seq data. If a vector of files is given, these are considered as replicates and are pooled together. Finally, noisy reads are discarded.

Usage

```
preprocess_bs_seq(files, file_format = "encode_rrbs", chr_discarded = NULL,
  min_bs_cov = 4, max_bs_cov = 1000)
```

Arguments

files	A vector of filenames containing replicate experiments. This can also be just a single replicate.
file_format	A string denoting the file format that the BS-Seq data are stored. Current version allows "encode_rrbs" or "bismark_cov" formats.
chr_discarded	A vector with chromosome names to be discarded.
min_bs_cov	The minimum number of reads mapping to each CpG site. CpGs with less reads will be considered as noise and will be discarded.
max_bs_cov	The maximum number of reads mapping to each CpG site. CpGs with more reads will be considered as noise and will be discarded.

Value

A GRanges object. The GRanges object contains two additional metadata columns:

- total_reads: total reads mapped to each genomic location.
- meth_reads: methylated reads mapped to each genomic location.

These columns can be accessed as follows: granges_object$total_reads

Additional Info

Information about the file formats can be found in the following links:

Encode RRBS format: http://rohsdb.cmb.usc.edu/GBshape/cgi-bin/hgTables?db=hg19&hgta_group=regulation&hgta_track=wgEncodeHaibMethylRrbs&hgta_table=wgEncodeHaibMethylRrbsBcbreas hgta_doSchema=describe+table+schema

Bismark Cov format: http://rnbeads.mpi-inf.mpg.de/data/RnBeads.pdf

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_bs_bismark_cov, read_bs_encode_haib pool_bs_seq_rep

Examples

```
# Obtain the path to the files
bs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
bs_data <- preprocess_bs_seq(bs_file, file_format = "encode_rrbs")

# Extract the total reads and methylated reads
total_reads <- bs_data$total_reads
meth_reads <- bs_data$meth_reads
```

---

preprocess_final_HTS_data

Pre-process final HTS data for downstream analysis

Description

preprocess_final_HTS_data performs a final filtering and preprocessing on the data for use in downstream analysis. These include, removing noisy gene expression data, removing or not unexpressed genes and log2-transorming of the FPKM values.

Usage

```
preprocess_final_HTS_data(methyl_region, prom_reg, rna_data,
  gene_log2_transf = TRUE, gene_outl_thresh = TRUE, gex_outlier = 300)
```

Arguments

methyl_region	Methylation region data, which are the output of the "create_methyl_region" function.
prom_reg	A GRanges object containing corresponding annotated promoter regions for each entry of the methyl_region list.
rna_data	A GRanges object containing corresponding RNA-Seq data for each entry of the methyl_region list. This is the output of the "read_rna_encode_caltech function.

gene_log2_transf
: Logical, whether or not to log2 transform the gene expression data.

gene_outl_thresh
: Logical, whehter or not to remove outlier gene expression data.

gex_outlier
: Numeric, denoting the threshold above of which the gene expression data (before the log2 transformation) are considered as noise.

Value

An object which contains following information:

- methyl_region: The subset of promoter methylation region data after the filtering process.
- gex: A vectoring storing only the corresponding gene expression values for each promoter region.
- rna_data: The corresponding gene expression data stored as a GRanges object.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_rna_encode_caltech process_haib_caltech_wrap

Examples

```
# Obtain the path to the BS file and then read it
bs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
bs_data <- read_bs_encode_haib(bs_file)

# Create promoter regions
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
annot_data <- read_rna_encode_caltech(rnaseq_file)
prom_region <- create_prom_region(annot_data)

# Create methylation regions
methyl_region <- create_methyl_region(bs_data, prom_region)

# Finally preprocess the HTS data
res <- preprocess_final_HTS_data(methyl_region, prom_region, annot_data)
```

---

process_haib_caltech_wrap

Wrapper method for processing ENCODE HAIB and Caltech HTS data

---

Description

process_haib_caltech_wrap is a wrapper method for processing HTS data and returning the methylation promoter regions and the corresponding gene expression data for those promoter regions. Note that the format of BS-Seq data should be in the Encode Haib bed format and for the RNA-Seq data in Encode Caltech bed format.

Usage

```
process_haib_caltech_wrap(bs_files, rna_files, chrom_size_file = NULL,
  chr_discarded = NULL, upstream = -7000, downstream = 7000,
  min_bs_cov = 4, max_bs_cov = 1000, cpg_density = 10, sd_thresh = 0.1,
  ignore_strand = TRUE, gene_log2_transf = TRUE, gene_outl_thresh = TRUE,
  gex_outlier = 300, fmin = -1, fmax = 1)
```

Arguments

bs_files	Filename (or vector of filenames if there are replicates) of the BS-Seq '.bed' formatted data to read values from.
rna_files	Filename of the RNA-Seq '.bed' formatted data to read values from. Currently, this version does not support pooling RNA-Seq replicates.
chrom_size_file	Optional filename containing genome chromosome sizes.
chr_discarded	A vector with chromosome names to be discarded.
upstream	Integer defining the length of bp upstream of TSS for creating the promoter region.
downstream	Integer defining the length of bp downstream of TSS for creating the promoter region.
min_bs_cov	The minimum number of reads mapping to each CpG site. CpGs with less reads will be considered as noise and will be discarded.
max_bs_cov	The maximum number of reads mapping to each CpG site. CpGs with more reads will be considered as noise and will be discarded.
cpg_density	Optional integer defining the minimum number of CpGs that have to be in a methylated region. Regions with less than n CpGs are discarded.
sd_thresh	Optional numeric defining the minimum standard deviation of the methylation change in a region. This is used to filter regions with no methylation change.
ignore_strand	Logical, whether or not to ignore strand information.
gene_log2_transf	Logical, whether or not to log2 transform the gene expression data.
gene_outl_thresh	Logical, whehter or not to remove outlier gene expression data.
gex_outlier	Numeric, denoting the threshold above of which the gene expression data (before the log2 transformation) are considered as noise.
fmin	Optional minimum range value for region location scaling. Under this version, this parameter should be left to its default value.
fmax	Optional maximum range value for region location scaling. Under this version, this parameter should be left to its default value.

Value

A processHTS object which contains following information:

- methyl_region: A list containing methylation data, where each entry in the list is an L_iX3 dimensional matrix, where L_i denotes the number of CpGs found in region i. The columns contain the following information:

1. 1st column: Contains the locations of CpGs relative to TSS. Note that the actual locations are scaled to the (fmin, fmax) region.
2. 2nd column: Contains the total reads of each CpG in the corresponding location.
3. 3rd column: Contains the methylated reads each CpG in the corresponding location.

- `prom_region`: A GRanges object containing corresponding annotated promoter regions for each entry of the `methyl_region` list. The GRanges object has one additional metadata column named `tss`, which stores the TSS of each promoter.
- `rna_data`: A GRanges object containing the corresponding RNA-Seq data for each entry of the `methyl_region` list. The GRanges object has three additional metadata columns which are explained in `read_rna_encode_caltech`
- `upstream`: Integer defining the length of bp upstream of TSS.
- `downstream`: Integer defining the length of bp downstream of TSS.
- `cpg_density`: Integer defining the minimum number of CpGs that have to be in a methylated region. Regions with less than `n` CpGs are discarded.
- `sd_thresh`: Numeric defining the minimum standard deviation of the methylation change in a region. This is used to filter regions with no methylation change.
- `fmin`: Minimum range value for region location scaling.
- `fmax`: Maximum range value for region location scaling.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

Examples

```
# Obtain the path to the files
rrbs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
proc_data <- process_haib_caltech_wrap(rrbs_file, rnaseq_file)
```

---

read_bs_bismark_cov *Read Bismark Cov formatted BS-Seq file*

Description

`read_bs_bismark_cov` reads a file containing methylation data from BS-Seq experiments using the `fread` function. The BS-Seq file should be in Bismark Cov format. Read the Important section below on when to use this function.

Usage

```
read_bs_bismark_cov(file, chr_discarded = NULL, is_GRanges = TRUE)
```

Arguments

`file`	The name of the file to read data values from.
`chr_discarded`	A vector with chromosome names to be discarded.
`is_GRanges`	Logical: if TRUE a GRanges object is returned, otherwise a data.frame object is returned.

Value

A GRanges object if is_GRanges is TRUE, otherwise a data.table object.

The GRanges object contains two additional metadata columns:

- total_reads: total reads mapped to each genomic location.
- meth_reads: methylated reads mapped to each genomic location.

These columns can be accessed as follows: granges_object$total_reads

Important

Unless you want to create a different workflow when processing the BS-Seq data, you should NOT call this function, since this is a helper function. Instead you should call the preprocess_bs_seq function.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

References

http://rnbeads.mpi-inf.mpg.de/data/RnBeads.pdf

See Also

pool_bs_seq_rep, preprocess_bs_seq

Examples

```
## Not run:
# Download the files and change the working directory to that location
file <- "name_of_bismark_file"
bs_data <- read_bs_bismark_cov(file)

# Extract the total reads and methylated reads
total_reads <- bs_data$total_reads
meth_reads <- bs_data$meth_reads

## End(Not run)
```

---

read_bs_encode_haib *Read ENCODE HAIB bed formatted BS-Seq file*

Description

read_bs_encode_haib reads a file containing methylation data from BS-Seq experiments using the scan function. The BS-Seq file should be in ENCODE HAIB bed format. Read the Important section below on when to use this function.

Usage

```
read_bs_encode_haib(file, chr_discarded = NULL, is_GRanges = TRUE)
```

Arguments

file	The name of the file to read data values from.
chr_discarded	A vector with chromosome names to be discarded.
is_GRanges	Logical: if TRUE a GRanges object is returned, otherwise a data.frame object is returned.

Value

A GRanges object if is_GRanges is TRUE, otherwise a data.table object.

The GRanges object contains two additional metadata columns:

- total_reads: total reads mapped to each genomic location.
- meth_reads: methylated reads mapped to each genomic location.

These columns can be accessed as follows: granges_object$total_reads

Important

Unless you want to create a different workflow when processing the BS-Seq data, you should NOT call this function, since this is a helper function. Instead you should call the preprocess_bs_seq function.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

References

http://rohsdb.cmb.usc.edu/GBshape/cgi-bin/hgTables?db=hg19&hgta_group=regulation&
hgta_track=wgEncodeHaibMethylRrbs&hgta_table=wgEncodeHaibMethylRrbsBcbreast0203015BiochainSitesR
hgta_doSchema=describe+table+schema

See Also

pool_bs_seq_rep, preprocess_bs_seq

Examples

```
# Obtain the path to the file and then read it
bs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
bs_data <- read_bs_encode_haib(bs_file)

# Extract the total reads and methylated reads
total_reads <- bs_data$total_reads
meth_reads <- bs_data$meth_reads
```

read_chrom_size *Read genome chromosome sizes file.*

Description

read_chrom_size reads a file containing genome chromosome sizes using the fread function.

Usage

```
read_chrom_size(file)
```

Arguments

file	The name of the file to read data values from.

Value

A data.table object.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_rna_encode_caltech, read_bs_encode_haib

Examples

```
chr_file <- system.file("extdata", "hg19.chr.sizes", package = "BPRMeth")
chr_data <- read_chrom_size(chr_file)

# Extract the size of the chr1
chr_data[1]
```

read_encode_cgi *Read file containing CpG island locations*

Description

read_encode_cgi reads a file containing CpG island (CGI) locations in the human genome using the fread function.

Usage

```
read_encode_cgi(file, is_GRanges = TRUE)
```

Arguments

file	The name of the file to read data values from.
is_GRanges	Logical: if TRUE a GRanges object is returned, otherwise a data.frame object is returned.

Value

A GRanges object if is_GRanges is TRUE, otherwise a data.table object.

The GRanges object contains one additional metadata column:

- cgi_id: Unique ID of the CpG Island.

This column can be accessed as follows: granges_object$cgi_id

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_rna_encode_caltech, read_bs_encode_haib

Examples

```
## Not run:
# Download the file and change the working directory to that location
file <- "name_of_CGI_file")
cgi_data <- read_encode_cgi(file)

# Extract the CGI ID
cgi_id <- cgi_data$cgi_id

## End(Not run)
```

---

read_rna_encode_caltech

Read ENCODE Caltech bed formatted RNA-Seq file

Description

read_rna_encode_caltech reads a file containing promoter annotation data together with gene expression levels from RNA-Seq experiments using the scan function. The RNA-Seq file should be in ENCODE Caltech bed format, e.g. use gtf2bed tool if your initial file is in gtf format.

Usage

```
read_rna_encode_caltech(file, chr_discarded = NULL, is_GRanges = TRUE)
```

Arguments

file	The name of the file to read data values from.
chr_discarded	A vector with chromosome names to be discarded.
is_GRanges	Logical: if TRUE a GRanges object is returned, otherwise a data.frame object is returned.

Value

A GRanges object if is_GRanges is TRUE, otherwise a data.table object.

The GRanges object contains three additional metadata columns:

- ensembl_id: Ensembl IDs of each gene promoter.
- gene_name: Gene name.
- gene_fpkm: Expression level in FPKM.

These columns can be accessed as follows: granges_object$ensembl_id

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

read_chrom_size, read_bs_encode_haib

Examples

```
# Obtain the path to the file and then read it
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
rna_data <- read_rna_encode_caltech(rnaseq_file)

# Extract the gene name and gene expression in fpkm
gene_name <- rna_data$gene_name
gene_fpkm <- rna_data$gene_fpkm
```

---

train_model_gex *Train gene expression model from methylation profiles*

Description

train_model_gex trains a regression model for predicting gene expression levels by taking as input the higher order methylation features extracted from specific genomic regions.

Usage

```
train_model_gex(formula = NULL, model_name = "svm", train,
  is_summary = TRUE)
```

Arguments

formula	An object of class formula, e.g. see lm function. If NULL, the simple linear regression model is used.
model_name	A string denoting the regression model. Currently, available models are: "svm", "randomForest", "rlm", "mars" and "lm".
train	The training data.
is_summary	Logical, print the summary statistics.

Value

A list containing the following elements:

- formula: The formula that was used.
- gex_model: The fitted model.
- train_pred The predicted values for the training data.
- train_errors: The training error metrics.

Author(s)

C.A.Kapourani <C.A.Kapourani@ed.ac.uk>

See Also

predict_model_gex

Examples

```
# Create synthetic data
train_data <- data.frame(x = rnorm(20), y=rnorm(20, 1, 4))
res <- train_model_gex(formula = y~., train = train_data)

# Using a different model
res <- train_model_gex(model_name = "randomForest", train = train_data)
```

Index