

Package ‘DMRforPairs’

October 27, 2015

Version 1.6.0

Date 2014-03-03

Title DMRforPairs: identifying Differentially Methylated Regions between unique samples using array based methylation profiles

Depends R (>= 2.15.2), Gviz (>= 1.2.1), R2HTML (>= 2.2.1), GenomicRanges (>= 1.10.7), parallel

Description DMRforPairs (formerly DMR2+) allows researchers to compare $n \geq 2$ unique samples with regard to their methylation profile. The (pairwise) comparison of n unique single samples distinguishes DMRforPairs from other existing pipelines as these often compare groups of samples in either single CpG locus or region based analysis. DMRforPairs defines regions of interest as genomic ranges with sufficient probes located in close proximity to each other. Probes in one region are optionally annotated to the same functional class(es). Differential methylation is evaluated by comparing the methylation values within each region between individual samples and (if the difference is sufficiently large), testing this difference formally for statistical significance.

License GPL (>= 2)

URL <http://www.martinrijlaarsdam.nl>,
<http://www.erasmusmc.nl/pathologie/research/lepo/3898639/>

biocViews Microarray, DNAMethylation, DifferentialMethylation, ReportWriting, Visualization, Annotation

Author Martin Rijlaarsdam [aut, cre], Yvonne vd Zwan [aut], Lambert Dorssers [aut], Leendert Looijenga [aut]

Maintainer Martin Rijlaarsdam <m.a.rijlaarsdam@gmail.com>

NeedsCompilation no

R topics documented:

DMRforPairs-package	2
calc_stats	4
CL.methy	5
DMRforPairs	6
example_output_DMRforPairs	9

export_data	10
merge_classes	12
plot_annotate_custom_region	13
plot_annotate_gene	15
plot_annotate_probes	16
plot_annotate_region	17
regionfinder	18
testregion	20
tune_parameters	21

Index	23
--------------	-----------

DMRforPairs-package *DMRforPairs package*

Description

Please see the DESCRIPTION file.

Note

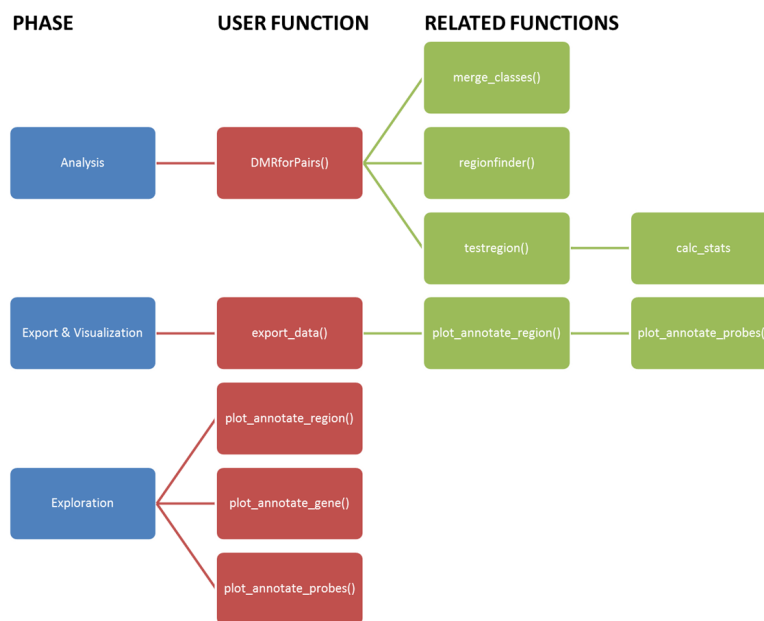
DMRforPairs is independent of the platform used to generate the data and the pipeline used to pre-process it. It can be used with any dataset as long as data frames are provided that have one matching row for each of the m probes. The examples provided in the vignette and associated publication focus mainly on data acquired using the Infinium HumanMethylation450 BeadChip (450K) platform. Methods to import and preprocess 450K data have been described extensively (see references below). Normalization and preprocessing are readily available in various pipelines and are not part of DMRforPairs. When assessing regions containing multiple probes, special care should be taken to normalize for probe type bias (for review, see Wilhelm-Benartzi et al 2013). In the following lists some general pointers are given on how to access the data for DMRforPairs using frequently used and publically available 450K pipelines. An example on how to process data from the CHARM platform is given in the supplementary data of the original DMRforPairs publication (Rijlaarsdam et al 2014, submitted).

- **lumi**: DMRforPairs is easily applied using 450K data imported by the lumi package (MethyLumiM object). The lumiMethyR() function can be used to read in a final report file as exported from GenomeStudio. The import includes intensity data and annotation information as exported by GenomeStudio. The lumi package offers extensive quality control and preprocessing options. The estimateBeta() and estimateM() functions should be used to extract beta and M values. The fData() method can be used to access annotation info. This manual focusses on the lumi package and supplies additional pointers on how to extract the necessary information using this package in the documentation of [DMRforPairs](#). For more information, please see the [LUMI Bioconductor page](#)
- **IMA**: the IMA package also uses the final report exported by GenomeStudio as the source for methylation data and annotation information (IMA.methy450R() function). Pre-processing is done via the IMA.methy450PP() function. beta values are readily available after import, while M values can be computed by calculating the logit2 of beta ($M = \log_2(\text{beta}/(1-\text{beta}))$). For more information, please see the [IMA homepage](#)

- `minfi`: `minfi` reads in the intensity information directly from the IDAT files, subsequently using the `read.450k.sheet()` and `read.450k.exp()` functions. `Minfi` also offers a range of QC and pre-processing options. the `getBeta()` and `getM()` functions are used to access the methylation quantities needed by `DMRforPairs`. Since `bioconductor 2.13`, this package depends on the annotation package `IlluminaHumanMethylation450kanno.ilmn12.hg19`. For more information, please see the [minfi](#) and [IlluminaHumanMethylation450kanno.ilmn12.hg19](#) Bioconductor pages.

Note

Writing permissions are required in the working directory to use the export and visualization functions of `DMRforPairs`. Internet access is required to use the annotation features. The figure below shows the relation between the functions described in this manual.



Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

References

- Bibikova, M., et al., *High density DNA methylation array with single CpG site resolution*. Genomics, 2011. 98(4): p. 288-95.
- Bibikova, M., et al., *Genome-wide DNA methylation profiling using Infinium(R) assay*. Epigenomics, 2009. 1(1): p. 177-200.
- Dedeurwaerder, S., et al., *Evaluation of the Infinium Methylation 450K technology*. Epigenomics, 2011. 3(6): p. 771-84.
- Du, P., W.A. Kibbe, and S.M. Lin, *lumi: a pipeline for processing Illumina microarray*. Bioinformatics, 2008. 24(13): p. 1547-8.
- Du, P., et al., *Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis*. BMC Bioinformatics, 2010. 11: p. 587.

- Maksimovic, J., L. Gordon, and A. Oshlack, *SWAN: Subset-quantile within array normalization for illumina infinium HumanMethylation450 BeadChips*. *Genome Biol*, 2012. 13(6): p. R44.
- Marabita, F., et al., *An evaluation of analysis pipelines for DNA methylation profiling using the Illumina HumanMethylation450 BeadChip platform*. *Epigenetics*, 2013. 8(3): p. 333-46.
- Pidsley, R., et al., *A data-driven approach to preprocessing Illumina 450K methylation array data*. *BMC Genomics*, 2013. 14(1): p. 293.
- Teschendorff, A.E., et al., *A beta-mixture quantile normalization method for correcting probe design bias in Illumina Infinium 450 k DNA methylation data*. *Bioinformatics*, 2013. 29(2): p. 189-96.
- Touleimat, N. and J. Tost, *Complete pipeline for Infinium((R)) Human Methylation 450K BeadChip data processing using subset quantile normalization for accurate DNA methylation estimation*. *Epigenomics*, 2012. 4(3): p. 325-41.
- Wang, D., et al., *IMA: an R package for high-throughput analysis of Illumina's 450K Infinium methylation data*. *Bioinformatics*, 2012. 28(5): p. 729-30.
- Wilhelm-Benartzi, C.S., et al., *Review of processing and analysis methods for DNA methylation array data*. *Br J Cancer*, 2013. 109(6): p. 1394-402.
- Eckhardt, F., et al., *DNA methylation profiling of human chromosomes 6, 20 and 22*. *Nat Genet*, 2006. 38(12): p. 1378-85.
- Rijlaarsdam, M.A., et al., *DMRforPairs: identifying Differentially Methylated Regions between unique samples using array based methylation profiles*. Submitted

 calc_stats

 CALCULATE DESCRIPTIVE AND TEST STATISTICS

Description

Calculate the (difference in) median beta and M values for the rows specified in probe_rows (descriptive statistics). probe_rows must specify a continuous range of adjacent probes. If a regions shows an high difference (>dM_min) between any pair of samples, the differences are formally tested (Mann Whitney U if n=2, Kruskal Wallis n>2). If n>2 and the omnibus test is significant , pairwise testing is performed to determine which differ significantly.

Usage

```
calc_stats(probe_rows, probes, m, b, n, dMth, do.format=FALSE)
```

Arguments

probe_rows	Vector of row numbers in regions\$valid.m, regions\$valid.beta or regions\$perprobe identifying adjacent probes. See output regionfinder
probes	see output of regionfinder (regions\$perprobe)
m	see output of regionfinder (regions\$valid.m)
b	see output of regionfinder (regions\$valid.beta)
n	total number of samples
dMth	see min_dM parameter of DMRforPairs
do.format	see testregion

Details

The p-values reported by this function are not yet adjusted for multiple testing. This function is rarely called by the user directly.

Value

Data frame or vector with statistics and test results concerning the probes specified in `probe_rows` (please see description).

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[DMRforPairs](#), [regionfinder](#), [testregion](#).

Examples

```
#calc_stats() is an integrated part of the DMRforPairs() wrapper and is
#not usually called by the user directly. Please see DMRforPairs() for
#an example.
```

CL.methy

METHYLATION DATA OF 4 CELL LINES (CHR 7)

Description

This dataset provides the average methylation values on chromosomes 7 (33-35MG, 105-108MB) of two commercially available EBV transfected lymphoblastoid cell lines from healthy individuals ([NA17105](#) (African American male) and [NA17018](#) (Chinese female), Coriell Institute for Medical Research (NJ, USA)). The dataset also contains the breast cancer cell line MCF7 (Soule et al. 1973) and HPV negative squamous-cell vulva carcinoma cell line A431 (Giard et al. 1973 & Hietanen et al. 1995). Raw data is publically available via the [Illumina website](#) and discussed in Bibikova et al. 2009 & 2011. Import and pre-processing was carried out using the LUMI package (Du et al. 2008) in R following the optimized "lumi: QN+BMIQ" pipeline (Marabita et al. 2013). This includes exclusion of poorly performing probes ($p < 0.01$, $n = 713$), color adjustment, quantile normalization and correction for probe type bias (Infinium I vs II) using the BMIQ algorithm (Teschendorf 2013).

Usage

```
CL.methy
```

Format

A data frame of 14 variables containing 711 observations. Variables: `targetID` (probe name), `chromosome`, `position (bp)`, `class.gene.related` (functional class related to gene), `class.island.related` (functional class related to CpG island), `gene.symbol`, `A431.M`, `MCF7.M`, `NA17105.M`, `NA17018.M`, `A431.beta`, `MCF7.beta`, `NA17105.beta`, `NA17018.beta`

Source

Bibikova et al 2009 & 2011, [download from Illumina website](#).

References

- Soule, H.D., et al., *A human cell line from a pleural effusion derived from a breast carcinoma*. J Natl Cancer Inst, 1973. 51(5): p. 1409-16.
- Giard, D.J., et al., *In vitro cultivation of human tumors: establishment of cell lines derived from a series of solid tumors*. J Natl Cancer Inst, 1973. 51(5): p. 1417-23.
- Hietanen, S., et al., *Human papillomavirus in vulvar and vaginal carcinoma cell lines*. Br J Cancer, 1995. 72(1): p. 134-9.
- Bibikova, M., et al., *High density DNA methylation array with single CpG site resolution*. Genomics, 2011. 98(4): p. 288-95.
- Bibikova, M., et al., *Genome-wide DNA methylation profiling using Infinium(R) assay*. Epigenomics, 2009. 1(1): p. 177-200.
- Du, P., W.A. Kibbe, and S.M. Lin, *lumi: a pipeline for processing Illumina microarray*. Bioinformatics, 2008. 24(13): p. 1547-8.
- Marabita, F., et al., *An evaluation of analysis pipelines for DNA methylation profiling using the Illumina HumanMethylation450 BeadChip platform*. Epigenetics, 2013. 8(3): p. 333-46.
- Teschendorff, A.E., et al., *A beta-mixture quantile normalization method for correcting probe design bias in Illumina Infinium 450k DNA methylation data*. Bioinformatics, 2013. 29(2): p. 189-96.

DMRforPairs

WRAPPER FOR THE DMRforPairs ANALYSIS

Description

Wrapper for the DMRforPairs analysis. Includes recoding of the probe classes, identification of sufficiently probe-dense regions and analysis (calculation of statistics and testing).

Usage

```
DMRforPairs(classes_gene, classes_island, targetID, chr, position, m.v, beta.v, min_n=4, min_distance=
```

Arguments

- `classes_gene` m x 1 data frame with the relation to gene for each probe. Illumina annotates this as Body, 5'UTR, 3'UTR, 1stExon, TSS1500 or TSS200. When the lumiMethyR function of the lumi package is used to import the data from GenomeStudio's final report, this information can be extracted via the fData(function), e.g. fData(data)\$UCSC_REFGENE_GROUP where data is the MethyLumiM object resulting from the import. If these classes are unknown an m x 1 character vector with "unknown.gene" can be supplied and recode should be set to 2.
- `classes_island` m x 1 data frame with the relation to CpG island. Illumina annotates this as Island, N_Shelf, N_Shore, S_Shelf or S_Shore. Information can be accessed via (fData(data)\$RELATION_TO_UCSC_CPG_ISLAND. If these classes are unknown an m x 1 character vector with "unknown.island" can be supplied and recode should be set to 2.

targetID	m x 1 data frame containing the identifier of each of the m probes, i.e. cgxxxxxxxx. Information can be accessed via <code>fData(data)\$TargetID</code>
chr	m x 1 data frame containing the chromosome each probe is annotated to. Information can be accessed via <code>fData(data)\$CHR</code>
position	m x 1 data frame containing the genomic position each probe is annotated to. Information can be accessed via <code>fData(data)\$MAPINFO</code>
m.v	m x n matrix or data frame containing the M-values for each probe for each sample. These can be directly extracted from the MethyLumiM using the <code>exprs(estimateM())</code> functions from the lumi package. Alternatively any m x n matrix can be used (for example output from an external normalization algorithm that does not use the MethyLumiM format or methylation values from another platform).
beta.v	See m.v. Use <code>exprs(estimateBeta(data))</code> to extract beta values from the MethyLumiM object.
min_n	Minimal number of probes required to consider a group of subsequent probes for inclusion as a region (=potential DMR). Default (minimum) is 4 as the Mann Whitney U test requires a minimum of 7 samples to ever reach a value < 0.05 (2x4=8).
min_distance	Maximal distance between adjacent probes to accept when including them in one region. The default value of 200bp is based on the findings in Marimba et al. 2013 and Eckhardt et al 2006 regarding rapid drop of co-methylation of adjacent probes when these are further apart.
min_dM	Minimal median difference between M-values in a region to consider the region for formal testing. Default value of 1.4 is based on the findings in Du et al 2010.
recode	Recoding scheme to group or discard probes annotated to certain functional genomic regions (see also <code>classes_gene</code> and <code>classes_island</code> parameters and the merge_classes function). (default=1)
sep	Separator used in the second column of the recode parameter. Use ";" or do not specify if using the built-in schemes. (default=";")
method	Method to use for correction for multiple testing. See <code>p.adjust()</code> function in R for possible settings. Default is 'fdr' implicating correction according to Benjamini-Hochberg.
debug.v	For development. If TRUE, only the first chromosome is analyzed. (default=FALSE)
gs	m x 1 data frame with associated gene symbols. From a MethyLumiM object this information can be extracted via the <code>fData(function)</code> , e.g. <code>fData(data)\$UCSC_REFGENE_NAME</code>
do.parallel	Use parallel processes to compute statistics per region. 0=no parallelization, -1=use all available cores, n>1 use n cores (default=0)

Details

This wrapper subsequently:

1. Recodes probe classes according to a custom or build in scheme. See `recode` parameter and [merge_classes](#).
2. Identifies regions with sufficient probe density (i.e. number of probes and proximity) over all genomic regions at which probes are annotated in the dataset. See [regionfinder](#)

- Calculates relevant statistics (e.g. median (difference in) M and beta values). If the median difference is sufficiently large ($\geq \text{min_dM}$), the function performs formal testing of the difference. If indicated, pairwise testing is performed as well. See [testregion](#) and [calc_stats](#).

Export & visualization are not directly included in the wrapper, as these require an internet connection and the analysis itself might take place on a compute server that is not connected to the internet. See [export_data](#) for more information about this. This wrapper does not store the results, but returns a list of data structures to be processed further or saved by the user.

Value

Returns a list of objects resulting from the different steps of the DMRforPairs algorithm. Please see the description of the associated functions for more information.

\$classes	includes the results from merge_classes
\$regions	includes the results from regionfinder
\$tested	includes the results from testregion

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

References

- Marabita, F., et al., *An evaluation of analysis pipelines for DNA methylation profiling using the Illumina HumanMethylation450 BeadChip platform*. Epigenetics, 2013. 8(3): p. 333-46.
- Eckhardt, F., et al., *DNA methylation profiling of human chromosomes 6, 20 and 22*. Nat Genet, 2006. 38(12): p. 1378-85.
- Du, P., et al., *Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis*. BMC Bioinformatics, 2010. 11: p. 587.

See Also

[merge_classes](#), [regionfinder](#), [testregion](#), [calc_stats](#).

Examples

```
#load vignette data & select a region of 4 MB on chr 7 for speed
data(DMRforPairs_data)
CL.methy=CL.methy[which(CL.methy$position<=1.07E+8 &
                      CL.methy$position>=1.06E+8),]

output=DMRforPairs(classes_gene=CL.methy$class.gene.related,
                  classes_island=CL.methy$class.island.related,
                  targetID=CL.methy$targetID,
                  chr=CL.methy$chromosome,
                  position=CL.methy$position,
                  m.v=CL.methy[,c(7:8)],
                  beta.v=CL.methy[,c(11:12)],gs=CL.methy$gene.symbol,
                  do.parallel=0)

#primary output of merge_classes()
head(output$classes$pclass) #original probe classes
head(output$classes$pclass_recoded) #recoded probe classes
head(output$classes$no.pclass) #row numbers of probes without a recoded class
```



```
head(output$classes$u_pclass) #classes used for recoding

#primary output of regionfinder()
#output listing the potential regions of interest identified
head(output$regions$boundaries)
#probes with associated class after recoding (=valid)
head(output$regions$valid.probes)
#... and associated m values for all samples
head(output$regions$valid.m)
#... and associated beta values for al samples
head(output$regions$valid.beta)
#matrix of valid probes (rows) and recoded probe classes (columns) with
#either NA if not included in any potential region of interest or the
#ID of the region the probe is assigned to.
head(output$regions$perprobe,10)

#primary output of testregion() and calc_stats()
#these results are similar to output$regions$boundaries
#but are supplemented with descriptive statistics
#and formal test results per region.
head(output$tested)
```

example_output_DMRforPairs

EXAMPLE OF DMRforPairs RESULTS

Description

This dataset includes the DMRforPairs results of a 4MB region in the CL.methy dataset. This is a small subset of the data used in the vignette to facilitate the examples in the reference manual (104MB-107MB on chr 7). Please see the CL.methy dataset for more details.

Usage

```
example_output_DMRforPairs
```

Format

DMRforPairs results formatted as described in the reference manual for the [DMRforPairs](#) wrapper.

Source

Bibikova et al 2009 & 2011, [download from Illumina website](#).

 export_data

 COMPLETE EXPORT AND VISUALIZATION OF DMRforPairs RESULTS

Description

Complete export of the DMRforPairs results. Export includes pdf's, thumbnails and TSV files describing the methylation status of all relevant (median delta M > dM_min) regions. M and beta values plotted against genomic position. Plots are generated for all samples and sample pairs (if n>2). Annotation information can be automatically looked up in the Ensembl database. If this is requested, integrated figures of the methylation pattern are generated including all transcripts annotated to the region of interest (+margin). This function also generates overviews (tables) in tsv and HTML format with information about the regions and their methylation status in the samples. This is done separately for all regions, regions with a relevant difference and regions with a significant difference.

Usage

```
export_data (tested,regions,th=0.05,annotate.relevant=FALSE,annotate.significant=TRUE,FigsNotRel
```

Arguments

tested	output of testregion
regions	output of regionfinder
th	p-value threshold for a DMR to be included as significant. (default=0.05; after correction for multiple testing)
annotate.relevant	By default annotations are only looked up for regions showing significant differential methylation patterns. Setting this parameter to TRUE will query Ensembl for the annotation info of each of the relevant (but not significant) regions as well. Please note that the database queries are slow and Ensembl might not accept an unlimited number of requests. (default=FALSE)
annotate.significant	By default annotations are only looked up for regions showing significant differential methylation patterns. Setting this parameter to FALSE will disable this feature (relevant on machines without an internet connection)(default=TRUE)
FigsNotRelevant	If TRUE, export_data() also exports figures (no annotation) for regions that do not show a relevant median difference in M value. Note that this will result in an extremely high number of files in the figures directory. (default=FALSE)
min_n	see DMRforPairs
min_dM	see DMRforPairs
min_distance	see DMRforPairs
margin	Margin in bp outside of each region that is queried in Ensembl and subsequently plotted). (default=10kb)
clr	Defines the colors to be used when plotting the different samples. If NA, colors are picked from the rainbow pallet. (default= NA)
method	see DMRforPairs

experiment.name	String. Is used to create a folder to export the results.
debug	see DMRforPairs

Details

In the output is organized according to all, relevant and significant regions:

1. all: all regions identified by DMRforPairs
2. relevant: regions identified by DMRforPairs with a median difference in M values between any pair of samples of > min_dM
3. significant: relevant + Kruskal Wallis test shows a significant result (<th)

Writing permissions are required in the working directory when using the export and visualization functions of DMRforPairs. Internet access is required to use the annotation features.

By default, DMRforPairs creates a folder (experiment.name) within the current working directory in which the output is stored. This is done within the [export_data](#) wrapper which should be run before any other (specific) analysis are performed (exploration phase, see [DMRforPairs-package](#)). A complete export generates a large number of files which are stored out of sight (experiment.name/figures). Figures and overviews of statistics are most conveniently accessed through the HTML tables.

Value

The output is comparable to that of [testregion](#), supplemented with external links (Ensembl, UCSC) and annotation info if requested.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[DMRforPairs](#), [regionfinder](#), [testregion](#), [plot_annotate_region](#), [plot_annotate_probes](#).

Examples

```
#For the this step, pleas ensure that you have writing
#permissions in your working directory. If any of the
#annotate parameters is set to TRUE, an internet connection
#is required.
data(DMRforPairs_data)
experiment="example_DMRforPairs"

#Export the results from output using the default settings;
#annotation is disabled as to prevent problems on (build)
#systems without an internet connection. A folder
#"example_DMRforPairs" containing the results is created in
#your working directory.
tested_inclannot=export_data(tested=example_output_DMRforPairs$tested,
regions=example_output_DMRforPairs$regions,
experiment=experiment,
annotate.significant=FALSE,
annotate.relevant=FALSE)
```

merge_classes

*MERGE ILLUMINA PROBE CLASSES***Description**

Illumina assigns quite specific functional classes to the probes. 11 classes are defined (Bibikova et al. 2009 & 2011):

- Relation to gene: *Body*, *5'UTR*, *3'UTR*, *1stExon*, *TSS1500*, *TSS200*
- Relation to CpG island: *Island*, *N_Shelf*, *N_Shore*, *S_Shelf*, *S_Shore*

As DMRforPairs looks at probes in close proximity to each other within each class, very specific annotations might result in too few probes per region per class and a subsequent drop in the number of identified regions and/or statistical power. This function therefore allows grouping and / or selection of classes of interest.

Usage

```
merge_classes(refgene_class, island_class, recode=1, sep=";")
```

Arguments

refgene_class	see description and classes_gene parameter in DMRforPairs
island_class	see description and classes_island parameter in DMRforPairs
recode	recoding scheme to use for the functional classes. Can be custom scheme (data frame) or build in scheme (0, 1 or 2). See details.
sep	Separator used in the second column of the recode parameter. Use ";" or do not specify if using the build in schemes.

Details

The recode parameter can be set to use one of the build in recoding schemes:

- 0: analyze all 11 classes annotated by Illumina separately
- 1: group *Body*, *5'UTR*, *3'UTR* into one category "gene" and *TSS1500* and *TSS200* in another ("tss"). All island associated classes are merged in one class "island"
- 2: all probes without subdivision into classes (also included probes associated with no class).

The recode parameter can also be set to a custom recoding scheme (data.frame). For example:

```
data.frame(c("gene", "tss"), c("Body;5'UTR;3'UTR;1stExon", "TSS1500;TSS200"))
```

In this scheme the classes are merged into two categories: TSS or other gene region. Probes solely associated with CpG island-related classes are discarded. Probes not annotated to any of the 11 classes are always discarded by DMRforPairs, except when option 2 is used which collects all probes into one class (i.e. ignores classes).

If classes are unknown an m x 1 character vector with "unknown.gene" and "unknown.island" for all m rows (probes) can be used for the refgene_class and island_class parameters respectively (set "recode" to 2!).

Value

List of objects containing:

\$pclass	original classes per probe (gene and island classes are merged) (m x 1 data frame)
\$pclass_merged	classes after recoding (m x 1 data frame)
\$no.pclass	row indexes of probes with no annotation to any of the classes specified in the recoding scheme.
\$u.pclass	unique list of the classes of interest after recoding (i.e. the first column of the recode data fram)

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

References

- Bibikova, M., et al., *High density DNA methylation array with single CpG site resolution*. Genomics, 2011. 98(4): p. 288-95.
- Bibikova, M., et al., *Genome-wide DNA methylation profiling using Infinium(R) assay*. Epigenomics, 2009. 1(1): p. 177-200.

Examples

```
#merge_classes() is an integrated part of the DMRforPairs() wrapper and is
#not usually called by the user directly. Please see DMRforPairs() for
#an example.
```

```
plot_annotate_custom_region
```

PLOT METHYLATION OF A CUSTOM GENOMIC REGION

Description

Wrapper for [plot_annotate_probes](#). Generates a plot of the methylation status of a custom genomic region.

Usage

```
plot_annotate_custom_region(chr,st,ed,regions,margin=10000,ID="CustomRegion",clr=NA,annotate=TRUE)
```

Arguments

chr	Chromosome on which the custom region is located. Number/X/Y only (i.e. no 'chr' prefix).
st	Start position (bp) of the custom region
ed	End position (bp) of the custom region
regions	See output of regionfinder
margin	See export_data

plot_annotate_gene *PLOT METHYLATION VALUES AROUND A GENE*

Description

Wrapper for [plot_annotate_probes](#). Generates a plot of the methylation status of / around a specific gene.

Usage

```
plot_annotate_gene(gs, regions, margin=10000, ID, clr=NA, annotate=TRUE, path)
```

Arguments

gs	Gene Symbol of interest
regions	Output of regionfinder
margin	See export_data
ID	Custom string which will be used as the file name of the figures generated
clr	See export_data
annotate	See plot_annotate_region
path	Path relative to the current working directory to save the files (=experiment.name in export_data)

Details

Expects a subfolder with the name of your experiment to exist as created by [export_data\(\)](#). Writing permissions are required in the working directory to use the export and visualization functions of [DMRforPairs](#). Internet access is required to use the annotation features.

Value

A number of figures (including pairwise plots if $n > 2$) is generated visualizing methylation status in the samples at / near the gene investigated. Statistics and optional annotation information are reported back by the function.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[export_data](#), [plot_annotate_probes](#), [regionfinder](#),

Examples

```
#For the this step, please ensure that you have writing
#permissions in your working directory. If any of the
#annotate parameters is set to TRUE, an internet connection
#is required.

#This example expects a folder "example_DMRforPairs" to exist and
#be writable in the current working directory.
data(DMRforPairs_data)
experiment="example_DMRforPairs"

plot_annotate_gene(gs="DUS4L",
                  regions=example_output_DMRforPairs$regions,
                  margin=10000,
                  ID="DUS4L",
                  annotate=FALSE,path=experiment)
```

plot_annotate_probes *PLOT METHYLATION+ANNOTATE A SET OF PROBES*

Description

Function that visualizes the methylation status of a selection of adjacent probes and (if requested) identifies transcripts associated with that region. If n>2 plots are generated for all samples together as well as in a pairwise fashion.

Usage

```
plot_annotate_probes(regions,title_x,probe_rows,margin=10000,ID=NA,clr=NA,annotate=TRUE,scores=N
```

Arguments

regions	See output of regionfinder .
title_x	String which will be used as the main title of the scatter plots of the beta and M values.
probe_rows	Row numbers of the adjacent probes of interest in the regions object. See output of regionfinder : \$perprobe, \$valid.probes, \$valid.m, \$valid.beta.
margin	see export_data
ID	String which will be used as the file name of the figures generated. (default = "CustomRegion")
clr	see export_data
annotate	see plot_annotate_region
scores	see plot_annotate_region
path	Path relative to the current working directory to save the files (=experiment.name in export_data)

Details

A plot will be generated using only probes present in the DMRforPairs output (i.e. associated to one or more of the recoded classes, see [merge_classes](#)). This function is rarely called directly by the user. There are a number of wrappers available which lookup the probes (probe_rows) associated with a regionID from DMRforPairs, a custom genomic region or a gene of interest and then call plot_annotate_probes().

Value

A number of figures (including pairwise plots if n>2) is generated visualizing methylation status in the samples at / near the probes investigated. Statistics and optional annotation information are reported back by the function.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[export_data](#), [plot_annotate_region](#), [regionfinder](#), [merge_classes](#).

Examples

```
#This function is called by the other plot_annotate...() wrappers and is
#not usually called by the user directly.
#Please see these functions for examples.
```

plot_annotate_region *PLOT METHYLATION AND ANNOTATE REGION*

Description

Wrapper for [plot_annotate_probes](#). Generates a plot of the methylation status of a region identified by DMRforPairs and associated transcripts (if annotate=TRUE).

Usage

```
plot_annotate_region(tested, regions, margin=10000, regionID, clr=NA, annotate=TRUE, scores=TRUE, path
```

Arguments

tested	Output of testregion
regions	Output of regionfinder
margin	See export_data
regionID	ID of a region identified by DMRforPairs as presented in the output of testregion and the output generated by export_data .
clr	See export_data
annotate	If FALSE, only basic visualizations without annotation information are generated. (default=TRUE)

scores	Specifies if statistics need to be calculated for the region or just figures. In a custom request from a user, statistics are relevant to calculate. In the <code>export_data</code> wrapper, they are redundant (already calculated in the <code>testregion</code> loop). (default=TRUE)
path	Path relative to the working directory to save the files (=experiment.name in <code>export_data</code>)

Details

Expects a subfolder with the name of your experiment to exist as created by `export_data()`. Writing permissions are required in the working directory to use the export and visualization functions of `DMRforPairs`. Internet access is required to use the annotation features.

Value

A number of figures (including pairwise plots if $n > 2$) is generated visualizing methylation status in the samples at / near the region investigated. Statistics and optional annotation information are reported back by the function.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[export_data](#), [plot_annotate_probes](#), [regionfinder](#), [testregion](#).

Examples

```
#For the this step, please ensure that you have writing
#permissions in your working directory. If any of the
#annotate parameters is set to TRUE, an internet connection
#is required.

#this function expects a folder "example_DMRforPairs" to exist and
#be writable in the current working directory.
data(DMRforPairs_data)
experiment="example_DMRforPairs"

plot_annotate_region(example_output_DMRforPairs$tested,
                     example_output_DMRforPairs$regions,
                     margin=10000,
                     regionID=7,
                     annotate=FALSE, scores=TRUE, path=experiment)
```

regionfinder

IDENTIFICATION OF PROBE DENSE REGIONS

Description

Identifies genomic regions with sufficient probe density, i.e. a high number of probes positioned closely together.

Usage

```
regionfinder(targetID, chr, position, pclass, r_excl, u_pclass, d=200, m.v, beta.v, n_min=4, debug=FALSE,
```

Arguments

targetID	See DMRforPairs parameters
chr	See DMRforPairs parameters
position	See DMRforPairs parameters
pclass	See merge_classes output
r_excl	See merge_classes output (<code>\$no.pclass</code>). If a probe is not annotated to one of the classes of interest in the recoding scheme, that probe is discarded.
u_pclass	See merge_classes output
d	See DMRforPairs parameters
m.v	See DMRforPairs parameters
beta.v	See DMRforPairs parameters
n_min	See DMRforPairs parameters
debug	See DMRforPairs parameters
gs	See DMRforPairs parameters

Details

Regions are identified (per unique functional class) based on a minimal number of probes that lay in close proximity to each other. This can lead to identical (start to end) regions in different classes due to overlapping annotations. `regionfinder()` merges such regions to prevent duplications. These regions can be identified by a combined class name (e.g. "tss;island").

Value

List of objects containing

<code>\$boundaries</code>	information about the regions identified (chr, start and end bp position, length of region, n of probes in region and information about the functional class(es) the region is associated to.
<code>\$perprobe</code>	m x p matrix with region IDs. Rows represent the valid probes (see below) and there is one column per functional class (see <code>u_pclass</code> parameter). Each probe can be associated to one region within each class. Region IDs are specific to a dataset and to a set of DMRforPairs parameters. They are not interchangeable between datasets/experiments and serve mainly as an identifier during exploration of the dataset.
<code>\$valid.m</code>	m x n matrix with M values for all samples. Only probes associated with at least one class of interest are included (=valid probes).
<code>\$valid.beta</code>	see <code>valid.m</code> . This matrix contains the beta values.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

[DMRforPairs](#), [merge_classes](#).

Examples

```
#regionfinder() is an integrated part of the DMRforPairs() wrapper and is
#not usually called by the user directly. Please see DMRforPairs() for
#an example.
```

testregion

METHYLATION DIFFERENECE & TESTING FOR 1 REGIONID

Description

Serves as a wrapper for `calc_stats`. `calc_stats` is a generic function using row numbers of adjacent probes which are to be queried for their difference in methylation levels between samples. `testregion()` identifies the row numbers of probes included in a specific regionID `x` and then calls `calc_stats`. It returns the difference in methylation level for region `x` and (if appropriate) the results of formal tests that evaluate this difference.

Usage

```
testregion(x, probes, m, b, n, dMth, do.format=FALSE)
```

Arguments

<code>x</code>	region ID, see output <code>regionfinder</code> .
<code>probes</code>	see output of <code>regionfinder</code> (<code>regions\$perprobe</code>).
<code>m</code>	see output of <code>regionfinder</code> (<code>regions\$valid.m</code>).
<code>b</code>	see output of <code>regionfinder</code> (<code>regions\$valid.beta</code>).
<code>n</code>	total number of samples.
<code>dMth</code>	see <code>min_dM</code> parameter of <code>DMRforPairs</code>
<code>do.format</code>	If TRUE, the function returns a formatted data frame with statistics regarding the region. If FALSE, it returns a basic vector. This is more computationally efficient if <code>testregion</code> is used in a loop (like in <code>DMRforPairs</code>).

Details

This function is rarely called by the user directly.

Value

Data frame or vector (depending on the setting of `do.format`) with statistics and test results concerning region `x`. For specification, please see `calc_stats`.

Author(s)

Martin Rijlaarsdam, <m.a.rijlaarsdam@gmail.com>

See Also

`calc_stats`

Examples

```
#testregion() is an integrated part of the DMRforPairs() wrapper and is
#not usually called by the user directly. Please see this function for
#an example.
```

tune_parameters	<i>TUNE MIN_DISTANCE AND MIN_N PARAMETERS</i>
-----------------	-----------------------------------------------

Description

This function evaluates the number of regions identified for various settings of `min_distance` and `n_min` (see [DMRforPairs](#)). It also summarizes the number of probes available inclusion using a specific setting of `recode` and the number of probes that is covered by the detected regions.

Usage

```
tune_parameters(parameters, classes_gene, classes_island, targetID, chr, position, m.v, beta.v, recode)
```

Arguments

<code>parameters</code>	A data frame with two columns: [1] <code>min_distance</code> and [2] <code>min_n</code> . The number of regions / probes will be evaluated for each pair of parameters (per row).
<code>classes_gene</code>	Please see DMRforPairs
<code>classes_island</code>	Please see DMRforPairs
<code>targetID</code>	Please see DMRforPairs
<code>chr</code>	Please see DMRforPairs
<code>position</code>	Please see DMRforPairs
<code>m.v</code>	Please see DMRforPairs
<code>beta.v</code>	Please see DMRforPairs
<code>recode</code>	Please see DMRforPairs . (Default=1)
<code>sep</code>	Please see DMRforPairs . (Default=";")
<code>gs</code>	Please see DMRforPairs
<code>do.parallel</code>	Tuning of the parameters requires a significant amount of compute time and power. Parallelization if offered to do this more efficiently. 0=no parallelization, -1=use all available cores, n>1 use n cores (default=0)

Details

This function requires a significant amount of compute time/power. Parallelization is highly recommended. When parallelization is enabled, no progress indicator is displayed by R.

Value

A data frame is returned with the following columns

<code>min_distance</code>	The setting of <code>min_distance</code> . Also see DMRforPairs
<code>min_n</code>	The setting for <code>min_n</code> . Also see DMRforPairs
<code>n.regions</code>	The number of regions identified in your data using the before mentioned combination of settings and the setting of <code>recode</code> .
<code>n.valid.probes</code>	The number of probes that were assigned to one or more classes in the merge_classes function. This is constant for a specific data set as long as the <code>recode</code> parameter is the same.
<code>n.probes.included</code>	The number of probes included in the identified regions.

Author(s)

Martin Rijlaarsdam

See Also

[regionfinder](#), [DMRforPairs](#)

Examples

```
data(DMRforPairs_data)
CL.methy=CL.methy[which(CL.methy$position<=1.07E+8 &
                      CL.methy$position>=1.06E+8),]
#just two relevant values for the parameters
#min_distance (100 & 300) and min_n (4 & 8)
parameters=expand.grid(min_distance = c(200), min_n = c(4,5))
results_parameters= tune_parameters(parameters,
classes_gene=CL.methy$class.gene.related,
classes_island=CL.methy$class.island.related,
targetID=CL.methy$targetID,
chr=CL.methy$chromosome,
position=CL.methy$position,
m.v=CL.methy[,c(7:8)],
beta.v=CL.methy[,c(11:12)],
recode=1,
gs=CL.methy$gene.symbol,
do.parallel=0)
results_parameters
```

Index

*Topic **analysis**

- calc_stats, [4](#)
- DMRforPairs, [6](#)
- merge_classes, [12](#)
- regionfinder, [18](#)
- testregion, [20](#)
- tune_parameters, [21](#)

*Topic **datasets**

- CL.methy, [5](#)
- example_output_DMRforPairs, [9](#)

*Topic **export_visualization**

- export_data, [10](#)
- plot_annotate_custom_region, [13](#)
- plot_annotate_gene, [15](#)
- plot_annotate_probes, [16](#)
- plot_annotate_region, [17](#)

calc_stats, [4](#), [8](#), [20](#)
CL.methy, [5](#)

DMRforPairs, [2](#), [4](#), [5](#), [6](#), [9–12](#), [14](#), [19–22](#)
DMRforPairs-package, [2](#)

example_output_DMRforPairs, [9](#)
export_data, [8](#), [10](#), [11](#), [13–18](#)

merge_classes, [7](#), [8](#), [12](#), [14](#), [17](#), [19](#), [22](#)

plot_annotate_custom_region, [13](#)
plot_annotate_gene, [15](#)
plot_annotate_probes, [11](#), [13–15](#), [16](#), [17](#),
[18](#)
plot_annotate_region, [11](#), [14–17](#), [17](#)

regionfinder, [4](#), [5](#), [7](#), [8](#), [10](#), [11](#), [13–18](#), [18](#),
[20](#), [22](#)

testregion, [4](#), [5](#), [8](#), [10](#), [11](#), [17](#), [18](#), [20](#)
tune_parameters, [21](#)