

Package ‘TileShuffle’

February 28, 2012

Type Package

Title Detection of differentially expressed segments in tiling array data

Version 0.2.0

Date 2011-04-08

Author Christian Otto, Kristin Reiche, Joerg Hackermueller

Depends R (>= 2.1.0)

Imports multtest, affxparser, intervals, genomeIntervals, utils, Biobase, limma

Suggests Starr

Maintainer Christian Otto <christian@bioinf.uni-leipzig.de>

Description This package contains functions for the analysis of tiling array data. It implements a statistical approach to detect expression or differential expression in terms of differences from the background distribution that avoids any intensity-related parameters. Moreover, it reduces the most dominant tiling array biases using an affinity-dependent permutation in conjunction with a windowing approach.

License GPL-2

LazyLoad yes

URL <http://www.bioinf.uni-leipzig.de/Software/TileShuffle/>

Collate ‘TileHelper.R’ ‘TileRead.R’ ‘TileShuffle.R’

R topics documented:

| | |
|--------------------------|---|
| TileReadCel | 2 |
| TileReadCustom | 4 |
| TileReadSignal | 6 |
| TileShuffle | 8 |

| | |
|--------------|-----------|
| Index | 14 |
|--------------|-----------|

 TileReadCel

TileReadCel

Description

Reads tiling array data from an Affymetrix BMAP file and CEL file(s).

Usage

```
TileReadCel( cel.filename, cel2.filename, celinc.filename,
             bmap.filename, minhits=8000, group="Hs", pmonly=TRUE,
             mmonly=FALSE, normalize=TRUE, mod.tstat=FALSE, gc=TRUE,
             matchscore=FALSE, verbose=FALSE)
```

Arguments

- `cel.filename` A vector of one or more filenames of Affymetrix CEL files (as character) that contain the probe intensities in the first cellular condition. Note that replicates are simply defined by more than one filename and used according to `mod.tstat`.
- `cel2.filename` A vector of one or more filenames of Affymetrix CEL files (as character) that contain the probe intensities in the second cellular condition. Note that replicates are simply defined by more than one filename and used according to `mod.tstat`.
- `celinc.filename` A vector of one or more filenames of Affymetrix CEL files (as character) containing probe intensities that should be included in the normalization. This may be desirable in case tiling array data of more than two different cellular states is available and multiple transitions between them are being analyzed.
- `bmap.filename` Filename of Affymetrix binary probe mapping (BMAP) file (as a character), which is a binary file containing information on the location of each probe in the reference sequence. Moreover, it stores the probe sequences that are necessary to calculate the GC content.
- `minhits` Minimal number of hits in BMAP entry to be considered for the further analysis. Due to historical reasons there are several entries in the BMAP file with only around thousand probes assigned that might overlap with the larger entries or with entries on other tiling arrays. In case of Affy tiling array 1.0R, a value of 8000 is recommended.
- `group` A group name as the organism abbreviation in order to consider only these entries in the BMAP file and hence disregard entries such as TIGR, Affymetrix, or bacterial controls.
- `pmonly` Indicates whether only intensities of perfect match (PM) probes on the Affymetrix tiling array are incorporated in the probe intensity estimation. If neither `pmonly` nor `mmonly` is set to TRUE, the specific hybridization effect of a probe is estimated by taking PM-MM. Due to the issue of higher intensities of mismatch (MM) probes compared to PM probes, it is recommended to enable this parameter.

| | |
|-------------------------|---|
| <code>mmonly</code> | Indicates whether only intensities of mismatch (MM) probes on the Affymetrix tiling array are incorporated in the probe intensity estimation. If neither <code>pmonly</code> nor <code>mmonly</code> is set to <code>TRUE</code> , the specific hybridization effect of a probe is estimated by taking <code>PM-MM</code> . This option is mutually exclusive with the <code>pmonly</code> parameter and is only recommended for investigating the behaviour of mismatch probes on a tiling array but not in common (differential) expression analyses. |
| <code>normalize</code> | Indicates whether the probe intensities of the given CEL files in <code>cel.filename</code> , <code>cel2.filename</code> , and <code>celinc.filename</code> are normalized by use of full-quantile normalization. The normalization is recommended if replicates are available or a differential analysis is executed and, hence, the transition between cellular states is analyzed. Note that both <code>PM</code> and <code>MM</code> probes are included in the normalization regardless of the <code>pmonly</code> or <code>mmonly</code> parameter. |
| <code>mod.tstat</code> | Indicates the use of replicate information. If <code>TRUE</code> , the score is the value of the moderated t-statistic (see <code>eBayes</code> of <code>limma</code> package for further details). Otherwise, the median probe <code>log2</code> -intensity among the given replicates or the median of all pairwise <code>log2</code> -fold changes between both states will be used as estimate of the probe differential score. Note that the moderated t-statistic can only be used if replicate information is available. |
| <code>gc</code> | Indicates whether GC content of probe sequences will be calculated. It is defined as fraction of both <code>G</code> s and <code>C</code> s in the probe sequence. |
| <code>matchscore</code> | Indicates whether match score will be read. The match score is defined as number of perfect matches of the probe sequence per megabase of the genomic sequence. This information needs to be set accordingly in the <code>BPMAP</code> file. Otherwise, all probes contain only the default value. |
| <code>verbose</code> | Indicates whether information on progress are printed. |

Details

Reads tiling array data in terms of an Affymetrix `BPMAP` (binary probe mapping) file and the Affymetrix `CEL` file that are created by the Affymetrix pipeline. The `CEL` file stores the results of the intensity calculations on the pixel values of the `DAT` file. This includes among others an intensity value, standard deviation of the intensity, and the coordinates of each `PM/MM` probe on the tiling array. These coordinate information can be merged with the Affymetrix `BPMAP` file that also comprises further information on the probes. This includes the genomic location of each probe in terms of the reference sequence and the probe center position, probe sequence for the `GC` calculation, and the corresponding probe sequence length. The method generates a `data.frame` comprising all required data on probes that are necessary for the subsequent shuffling analysis.

Value

Returns a `data.frame` containing information on all probes, i.e., the full name of the reference sequence (organism abbreviation and chromosome name), the chromosome name, the probe center position, the length of the probe sequence, the `GC` content of the probe sequence, the match score (if `matchscore` is enabled), and the probe score.

Examples

```
## Note that the following example only executes if the external data
## of the Starr R package is available which includes an artificial
## Affymetrix BPMAP file and corresponding CEL files.
path <- system.file("extdata", package = "Starr")
if (path != ""){
```

```

## define Affymetrix BMAP file for probe mapping
bmap.filename <- file.path(path, "Scerevisiae_tlg_chr1.bmap")
## define Affymetrix CEL files
## here: file of control experiment (wt)
wt.filename <- file.path(path, "wt_IP_chr1.cel")
stopifnot(file.exists(bmap.filename) && file.exists(wt.filename))

## read CEL file and return data.frame
## with information such as genomic localization
## of probes, GC content of probe sequences,
## and probe intensities.
## Note that group is '' (blank) for old Affy chr21/22arrays
## while it is "Hs" for other Human, "Mm" for Mouse, or "Dm"
## for Drosophila tiling array platforms.
wt.cel <- TileReadCel(
  cel.filename=wt.filename,
  bmap.filename=bmap.filename,
  group="", gc=TRUE, verbose=FALSE)

## getting an overview on the reported data.frame
str(wt.cel)

## investigating data
## e.g. plot density of intensities
pdf(file="wt_int_density.pdf")
plot(density(wt.cel$intensity), main="", xlab="Intensity")
dev.off()
## or GC bias with three GC content bins
pdf(file="wt_gc_bias.pdf")
boxplot(wt.cel$intensity ~ cut(wt.cel$gc,
  breaks=c(0,0.36,0.52,Inf),right=FALSE),
  xlab="GC content", ylab="Intensity")
dev.off()
## cleanup
rm(bmap.filename, wt.filename, wt.cel)
}
rm(path)

```

TileReadCustom

TileReadCustom

Description

Reads tiling array data from custom-formatted files.

Usage

```

TileReadCustom(custom.filename, custom2.filename, custominc.filename,
  pmonly=TRUE, mmonly=FALSE, normalize=TRUE, mod.tstat=FALSE,
  gc=TRUE, verbose=FALSE)

```

Arguments

custom.filename

A vector of one or more filenames of custom- formatted files (as character) that contain the probe intensities in the first cellular condition and are formatted

as described. Note that replicates are simply defined by more than one filename and used according to `mod.tstat`.

| | |
|---------------------------------|---|
| <code>custom2.filename</code> | A vector of one or more filenames of custom-formatted files (as character) that contain the probe intensities in the second cellular condition and are formatted as described. Note that replicates are simply defined by more than one one filename and used according to <code>mod.tstat</code> . |
| <code>custominc.filename</code> | A vector of one or more filenames of custom-formatted files (as character) containing probe intensities that should be included in the normalization. This may be desirable in case tiling array data of more than two different cellular states is available and multiple transitions between them are being analyzed. |
| <code>pmonly</code> | Indicates whether only intensities of perfect match (PM) probes on the tiling array are incorporated in the probe intensity estimation. If neither <code>pmonly</code> nor <code>mmonly</code> is set to TRUE, the specific hybridization effect of a probe is estimated by taking PM-MM. |
| <code>mmonly</code> | Indicates whether only intensities of mismatch (MM) probes are incorporated in the probe intensity estimation. If neither <code>pmonly</code> nor <code>mmonly</code> is set to TRUE, the specific hybridization effect of a probe is estimated by taking PM-MM. This option is mutually exclusive with the <code>pmonly</code> parameter and is only recommended for investigating the behaviour of mismatch probes but not in common (differential) expression analysis. |
| <code>normalize</code> | Indicates whether the probe intensities of the given files in <code>custom.filename</code> , <code>custom2.filename</code> , and <code>custominc.filename</code> are normalized by use of full-quantile normalization. The normalization is recommended if replicates are available or a differential analysis is executed and, hence, the transition between cellular states is analyzed. Note that PM and MM probe intensities are not included in the normalization if <code>mmonly</code> and <code>pmonly</code> is set to TRUE, respectively. |
| <code>mod.tstat</code> | Indicates the use of replicate information. If TRUE, the score is the value of the moderated t-statistic (see <code>eBayes</code> of <code>limma</code> package for further details). Otherwise, the median probe \log_2 -intensity among the given replicates or the median of all pairwise \log_2 -fold changes between both states will be used as estimate of the probe differential score. Note that the moderated t-statistic can only be used if replicate information is available. |
| <code>gc</code> | Indicates whether GC content of probe sequences will be calculated. It is defined as fraction of both Gs and Cs in the probe sequence. The probe sequences may be set arbitrarily if <code>gc</code> is disabled. |
| <code>verbose</code> | Indicates whether information on progress are printed. |

Details

Reads tiling array data from custom-formatted files that may be created by any tiling array platform. It includes probe information that is separated by tabulators. Except for comment (indicated by '#' on beginning) or empty line, each line must contain the following columns: probe identifier that is unique within each file and consistent among the given files in terms of their probe coordinates, name of the reference sequence, start and end position of the probe on the reference sequence (both 0-based, as integer), intensity value of PM probe (non-log scale), intensity value of MM probe (on non-log scale), and probe sequence in order to calculate the GC content. The data must not contain 'NA' values. Moreover, the probe sequences may be set arbitrary if `gc` is disabled. The

method generates a `data.frame` comprising all required data on probes that are necessary for the subsequent shuffling analysis.

Value

Returns a `data.frame` containing information on all probes, i.e., the name of the reference sequence, the chromosome name (here: both are equal), the probe center position, the length of the probe sequence, the GC content of the probe sequence, the match score (if `matchscore` is enabled), and the \log_2 probe score.

Examples

```
## This example requires the custom-formatted files
## in the extdata folder of this package. Otherwise,
## it aborts with an error.
path <- system.file("extdata", package = "TileShuffle")
stopifnot(path != "")
## define filename to custom-formatted file
custom.filename <- file.path(path, "custom.txt")
stopifnot(file.exists(custom.filename))

## read custom-formatted file and return data.frame
## with information such as genomic localization
## of probes, GC content of probe sequences, and
## probe intensities.
custom <- TileReadCustom(custom.filename=custom.filename,
pmonly=TRUE, gc=TRUE, verbose=FALSE)

## getting an overview on the reported data.frame
str(custom)

## investigating data
## e.g. plot density of intensities
pdf(file="custom_int_density.pdf")
plot(density(custom$intensity), main="", xlab="Intensity")
dev.off()
## or GC bias with three GC content bins
pdf(file="custom_gc_bias.pdf")
boxplot(custom$intensity ~ cut(custom$gc,
breaks=c(0,0.36,0.52,Inf),right=FALSE),
xlab="GC content", ylab="Intensity")
dev.off()
## cleanup
rm(path, custom.filename, custom)
```

TileReadSignal

TileReadSignal

Description

Reads tiling array data from an Affymetrix BMAP file and a TAS-created signal.txt file.

Usage

```
TileReadSignal(signal.filename, bmap.filename, minhits=8000,
              group="Hs", gc=TRUE, matchscore=FALSE, verbose=FALSE)
```

Arguments

| | |
|-----------------|--|
| signal.filename | Filename of human-readable signal.txt file (as a character) created by the Tiling Array Software (TAS). |
| bmap.filename | Filename of Affymetrix binary probe mapping (BMAP) file (as a character), which is a binary file containing information on the location of each probe in the reference sequence. Moreover, it stores the probe sequences that are necessary to calculate the GC content. |
| minhits | Minimal number of hits in BMAP entry to be considered for the further analysis. Due to historical reasons there are several entries in the BMAP file with only around thousand probes assigned that might overlap with the larger entries or with entries on other tiling arrays. In case of Affy tiling array 1.0R, a value of 8000 is recommended. |
| group | A group name as the organism abbreviation in order to consider only these entries in the BMAP file and hence disregard entries such as TIGR, Affymetrix, or bacterial controls. |
| gc | Indicates whether GC content of probe sequences will be calculated. It is defined as fraction of both Gs and Cs in the probe sequence. |
| matchscore | Indicates whether match score will be read. The match score is defined as number of perfect matches of the probe sequence per megabase of the genomic sequence. This information needs to be set accordingly in the BMAP file. Otherwise, all probes contain only the default value. |
| verbose | Indicates whether information on progress are printed. |

Details

Reads tiling array data in terms of an Affymetrix BMAP (binary probe mapping) file and the human-readable signal.txt files that are created by the Tiling Array Software (TAS) from Affymetrix. The signal files comprise information on the genomic map entry (e.g. chromosome name), probe center positions and probe intensities. The intensities are assumed to be in \log_2 scale. The method generates a `data.frame` comprising all required data on probes that are necessary for the subsequent shuffling analysis.

Value

Returns a `data.frame` containing information on all probes, i.e., the full name of the reference sequence (organism abbreviation and chromosome name), the chromosome name, the probe center position, the length of the probe sequence, the GC content of the probe sequence, the match score (if `matchscore` is enabled), and the probe score.

Note

This method is deprecated since it was specifically designed for the Affymetrix Human tiling array 1.0R platform and the files created by TAS. Please consider using [TileReadCel](#) or [TileReadCustom](#) instead.

 TileShuffle

TileShuffle

Description

The overall procedure to analyze tiling array expression data.

Usage

```
TileShuffle(input.type, signal.filename, cel.filename, cel2.filename,
            celinc.filename, custom.filename, custom2.filename,
            custominc.filename, bmap.filename, minhits=8000, group="Hs",
            pmonly=TRUE, normalize=TRUE, mod.tstat=FALSE, noofperms, winsize,
            qvalue, gcmode="fixed", gcnun, score.function="trimmed",
            randomize=FALSE, diff, diff.variant="B", regions.filename,
            zscore.filename, output.filename, verbose=FALSE)
```

Arguments

- `input.type` The type of input data which contains the probe and intensity information. In case of `input.type == 1`, the input is given in terms of human-readable `signal.txt` files, created by the Affymetrix Tiling Array Software (TAS) that contain the probe scores, and Affymetrix BMAP files comprising the probe mapping and further information on them. In case of `input.type == 2`, the input is provided as Affymetrix CEL files and Affymetrix BMAP files. On the other hand, by selecting `input.type == 3`, the input is provided as custom-formatted files that may be created from tiling array data of any design and platform. More details on the format are given in the description of the function [TileReadCustom](#). The selection of `input.type == 1` is deprecated since it was specifically designed for the Affymetrix pipeline with TAS on the Human tiling array 1.0R platform and may not work properly in other cases. Instead, please use `input.type = 2` with Affymetrix CEL files or `input.type = 3` with custom-formatted text files.
- `signal.filename` Filename of human-readable `signal.txt` file (as a `character`) created by the Tiling Array Software (TAS).
- `cel.filename` A vector of one or more filenames of Affymetrix CEL files (as `character`) that contain the probe intensities in the first cellular condition. Note that replicates are simply defined by more than one filename and used according to `mod.tstat`.
- `cel2.filename` A vector of one or more filenames of Affymetrix CEL files (as `character`) that contain the probe intensities in the second cellular condition. Note that replicates are simply defined by more than one filename and used according to `mod.tstat`.
- `celinc.filename` A vector of one or more filenames of Affymetrix CEL files (as `character`) containing probe intensities that should be included in the normalization. This may be desirable in case tiling array data of more than two different cellular states is available and multiple transitions between them are being analyzed.

| | |
|---------------------------------|---|
| <code>custom.filename</code> | A vector of one or more filenames of custom- formatted files (as character) that contain the probe intensities in the first cellular condition and are formatted as described. Note that replicates are simply defined by more than one filename and used according to <code>mod.tstat</code> . |
| <code>custom2.filename</code> | A vector of one or more filenames of custom- formatted files (as character) that contain the probe intensities in the second cellular condition and are formatted as described. Note that replicates are simply defined by more than one one filename and used according to <code>mod.tstat</code> . |
| <code>custominc.filename</code> | A vector of one or more filenames of custom-formatted files (as character) containing probe intensities that should be included in the normalization. This may be desirable in case tiling array data of more than two different cellular states is available and multiple transitions between them are being analyzed. |
| <code>bpmap.filename</code> | Filename of Affymetrix binary probe mapping (BPMAP) file (as a character), which is a binary file containing information on the location of each probe in the reference sequence. Moreover, it stores the probe sequences that are necessary to calculate the GC content. |
| <code>minhits</code> | Minimal number of hits in BPMAP entry to be considered for the further analysis. Due to historical reasons there are several entries in the BPMAP file with only around thousand probes assigned that might overlap with the larger entries or with entries on other tiling arrays. In case of Affy tiling array 1.0R, a value of 8000 is recommended. |
| <code>group</code> | A group name as the organism abbreviation in order to consider only these entries in the BPMAP file and hence disregard entries such as TIGR, Affymetrix, or bacterial controls. |
| <code>pmonly</code> | Indicates whether only intensities of perfect match (PM) probes on the tiling array are incorporated in the probe intensity estimation. If neither <code>pmonly</code> nor <code>mmonly</code> is set to <code>TRUE</code> , the specific hybridization effect of a probe is estimated by taking PM-MM. |
| <code>normalize</code> | Indicates whether the probe intensities of the given files in <code>cel.filename</code> , <code>cel2.filename</code> , and <code>celinc.filename</code> (with <code>input.type == 2</code> or <code>custom.filename</code> , <code>custom2.filename</code> , and <code>custominc.filename</code> (with <code>input.type == 3</code>) are normalized by use of full-quantile normalization. The normalization is recommended if replicates are available or a differential analysis is executed and, hence, the transition between cellular states is analyzed. |
| <code>mod.tstat</code> | Indicates the use of replicate information. If <code>TRUE</code> , the score is the value of the moderated t-stastistic (see <code>eBayes</code> of <code>limma</code> package for further details). Otherwise, the median probe <code>log2</code> -intensity among the given replicates or the median of all pairwise <code>log2</code> -fold changes between both states will be used as estimate of the probe differential score. Note that the moderated t-statistic can only be used if replicate information is available. |
| <code>noofperms</code> | Number of permutations used to sample the background distribution. With higher number of permutations, the statistical significance of windows can be assessed more precisely in particular with more restrictive significance thresholds, i.e., low values of the <code>qvalue</code> parameter. Moreover, in case of the differential analysis (<code>diff</code> is enabled), the number of permutations should be increased to |

| | |
|-----------------------------|---|
| | sample the background distribution more accurately. In general, <code>noofperm</code> is recommended to be set to 10000 and 100000 in case of expression and differential expression analyses, respectively. |
| <code>winsize</code> | Maximal width of the windows that are being statistically assessed. The width is defined as the difference in the genomic center positions of the first and last enclosed probe. Due to gaps in the commonly uniform distribution of probes over the genomic sequence. The analyzed windows may be considerably shorter than the defined <code>winsize</code> or may even consist of only one probe. |
| <code>qvalue</code> | Maximal permitted q-value that is applied in the statistical analysis. Hence, windows with a q-value above the given value will not be included in the returned <code>data.frame</code> . |
| <code>gcmode</code> | Mode of GC content binning. In case of <code>gcmode</code> set to "fixed", the classification of probes in bins was predefined considering the GC content effect on probe intensities on the Affymetrix tiling array 1.0R platform. In this case, only the following values are permitted: 1, 2, 3, 4, or 5. By setting <code>gcmode</code> to "automatic", the binning is done automatically solely on the distribution of the GC content of the probes in order to obtain GC content bins that are optimally balanced in terms of their sizes. |
| <code>gcnum</code> | Number of different GC content bins where probes within each bin have a similar expected sequence-specific affinity and are permuted independently from each other. Accordingly, intensities of probes that belong to different affinity bins must not be interchanged. Due to the trade-off between the reduction of the sequence-specific effect and the maintenance of sufficiently large permutation bins, three GC content bins are recommended in the expression analysis. In case of <code>gcmode</code> set to "fixed", the classification of probes in bins was predefined considering the GC content effect on probe intensities on the Affymetrix tiling array 1.0R platform. In this case, only the following values are permitted: 1, 2, 3, 4, or 5. By setting <code>gcmode</code> to "automatic", the binning is done automatically solely on the distribution of the GC content of the probes in order to obtain GC content bins that are optimally balanced in terms of their sizes. Note that the <code>gcnum</code> is set to one in case of differential expression analysis (<code>diff</code> enabled) since sequence-specific effect cancel out and affinity binning is rendered unnecessary. |
| <code>score.function</code> | Function to calculate windows scores over the scores of the corresponding probes, i.e., arithmetic average (<code>score.function = "mean"</code>), arithmetic mean trimmed by the minimal and maximal value (<code>score.function = "trimmed"</code>), or the median (<code>score.function = "median"</code>). Note that the definition of trimmed mean differs from the common one with given percentile ranges. Moreover, the resulting scores with trimmed mean may differ from the mean only in case of windows that contain more than two probes. The latter two scoring functions are recommended due to their higher robustness against outliers. However, due to the higher calculation costs, the running time increases by selecting "trimmed" or "median". Note that the function is given as <code>character</code> . |
| <code>randomize</code> | Indicates whether an additional permutation is applied prior to the calculation of original window scores. It is a possibility to roughly estimate the false positive rate since under the assumption of mostly unexpressed probes no window over permuted intensities is expected to differ significantly from the background distribution. |
| <code>diff</code> | Indicates whether differential expression analysis is applied. |

`diff.variant` The variants of the differential expression analysis differ in score calculation, in the permutation procedure as well as in their assignment of statistical significance to windows. The `diff.variant A` is similar to the normal expression analysis but two-tailed p-values are estimated to regard both regulation directions, up and down. The multiple testing correction is then adjusted to account for these additional comparisons. The `diff.variant B` assumes that entire windows are either up- or down-regulated between conditions. The presumed direction of regulation is initially assigned to each window on the basis of its score. Subsequently, all converse probes, i.e., probes with negative score within positive windows or vice versa, are ignored and neither permuted nor incorporated into the score calculation. Consequently, positive and negative windows are compared to different background distributions. The p-value estimation and correction is done equivalent as in the case of the normal expression analyses. Both variants produce fairly similar results while the variant B is slightly superior in its performance and hence recommended.

`regions.filename`

Filename of BED-formatted file that contains regions to which the (differential) expression analysis should be limited to. Hence, only windows entirely enclosed in the union of these regions are statistically evaluated. Commonly, this parameter is used in order to identify highly and differentially expression (highdiff) regions by restricting the differential expression analysis (`diff` enabled) to regions identified as highly expressed in either one of the corresponding cellular conditions.

`zscore.filename`

Filename of BED-formatted z-score file that may be written and comprises the analyzed windows of the statistical analysis. Note that it is only written if `zscore.filename` is set accordingly. In this case, it contains the name of the reference sequence, start and end position, description, estimate z-score, and '+' as strand for each analyzed window. The description is an underscore-delimited string of the number of covered probes, the average GC content of their sequences, the window q-value multiplied by 100, and the window score calculated with the given `scoring.function` on the probe scores. The z-scores represent normalized window scores that are calculated by use of the sampled background. More precisely, the window z-score z is calculated by $z = \frac{x-\mu}{\sigma}$ where x is the window score and μ and σ are the mean and standard deviation of the permuted window scores, respectively. Note that negative z-scores indicate down-regulation and positive indicate up-regulated regions since z-scores are bounded to zero from below or from above in case of up- or down-regulation, respectively. Hence, using common expression analysis (if `diff` is FALSE), the z-score cannot be negative. Note that any existing file will be overwritten.

`output.filename`

Filename of BED-formatted output file that is written and comprises information on the segments identified as significantly (differentially) expressed including name of reference sequence, start and end position, description, score that is uniformly set to '10' (better compatibility of BED files e.g. with UCSC genome browser), and '+' as strand. The description is an underscore-delimited string of the name of reference sequence, the start and end position, the number of probes covered by the segment, the average GC content of their sequences, the minimal q-value of windows that were merged into the segment (multiplied by 100), and the segment score calculated with the given `scoring.function` on the covered probes.

verbose Indicates whether information on progress are printed.

Details

The overall procedure to analyze tiling array expression data by reading the measure probe scores and further information as input, executes the statistical analysis of this data, and report the output in terms of one or two BED-formatted files (depending on whether `zscore.filename` is set to NULL).

Value

None. All output is written to the `output.filename` and `zscore.filename` if `zscore.filename` is not NULL. All generated files contain header information marked by '#' at the beginning of the line that lists the used parameters with their values and the description of the columns in the subsequent output. Note that any existing file will be overwritten.

Examples

```
## Note that the following example only executes if the external data
## of the Starr R package is available which includes an artificial
## Affymetrix BMAP file and corresponding CEL files.
path <- system.file("extdata", package = "Starr")
if (path != ""){
  ## define Affymetrix BMAP file for probe mapping
  bmap.filename <- file.path(path, "Scerevisiae_tlg_chr1.bmap")
  ## define Affymetrix CEL files
  ## here: file of control experiment (wt)
  wt.filename <- file.path(path, "wt_IP_chr1.cel")
  ## and files of real experiment with
  ## tagged Rpb3 with two replicates (ip)
  ip.filename <- c(file.path(path, "Rpb3_IP_chr1.cel"),
  file.path(path, "Rpb3_IP2_chr1.cel"))
  stopifnot(file.exists(bmap.filename) && all(file.exists(ip.filename)) &&
  file.exists(wt.filename))

  ## identify highly expressed segments in IP state
  ## (only 100 permutations as example)
  ## Note that group is here '' (blank) for old Affy chr21/22 arrays
  ## but commonly it is "Hs" for Human, "Mm" for Mouse or "Dm" for Drosophila
  TileShuffle(bmap.filename=bmap.filename, cel.filename=ip.filename,
  input.type=2, group="", ponly=TRUE, normalize=TRUE,
  noofperms=100, winsize=200, qvalue=0.05,
  gcnum=3, diff=FALSE, output.filename="ip_high.bed",
  zscore.filename="ip_high_zscore.bed", verbose=FALSE)

  ## identify highly expressed segments in wt state
  ## (only 100 permutations as example)
  TileShuffle(bmap.filename=bmap.filename, cel.filename=wt.filename,
  input.type=2, group="", ponly=TRUE, normalize=TRUE,
  noofperms=100, winsize=200, qvalue=0.05,
  gcnum=3, diff=FALSE, output.filename="wt_high.bed",
  zscore.filename="wt_high_zscore.bed", verbose=FALSE)

  ## concatenate files
  file.create("ip_and_wt_high.bed")
  file.append("ip_and_wt_high.bed", "ip_high.bed")
}
```

```
file.append("ip_and_wt_high.bed", "wt_high.bed")

## identify highly and differentially expressed segments (highdiff)
## that are highly expressed in control or real experiment and
## significantly differentially expressed between both conditions
## (only 100 permutations as example)
## Note: common differential analysis without identifying highdiff segments
## is executed simply by omitting the parameter 'regions.filename'
TileShuffle(bpmap.filename=bpmap.filename, cel.filename=wt.filename,
cel2.filename=ip.filename, input.type=2, group="",
pmonly=TRUE, normalize=TRUE, noofperms=100,
winsize=200,qvalue=0.05, gcnum=1, diff=TRUE,
regions.filename="ip_and_wt_high.bed",
output.filename="ip_and_wt_highdiff.bed",
zscore.filename="ip_and_wt_highdiff_zscore.bed",
verbose=FALSE)
## cleanup
rm(bpmap.filename, wt.filename, ip.filename)
}
rm(path)
```

Index

TileReadCel, [2](#), [7](#)
TileReadCustom, [4](#), [7](#), [8](#)
TileReadSignal, [6](#)
TileShuffle, [8](#)