Package ‘skewr’

March 29, 2024

Title Visualize Intensities Produced by Illumina's Human Methylation 450k BeadChip

Version 1.35.0

Description The skewr package is a tool for visualizing the output of the Illumina Human Methylation 450k BeadChip to aid in quality control. It creates a panel of nine plots. Six of the plots represent the density of either the methylated intensity or the unmethylated intensity given by one of three subsets of the 485,577 total probes. These subsets include Type I-red, Type I-green, and Type II. The remaining three distributions give the density of the Beta-values for these same three subsets. Each of the nine plots optionally displays the distributions of the `rs` SNP probes and the probes associated with imprinted genes as series of 'tick' marks located above the x-axis.

Depends R (>= 3.1.1), methylumi, wateRmelon, mixsmsn, IlluminaHumanMethylation450kmanifest

Imports minfi, S4Vectors (>= 0.19.1), RColorBrewer

Suggests GEOquery, knitr, minfiData

VignetteBuilder knitr

License GPL-2

LazyData true

biocViews DNAMethylation, TwoChannel, Preprocessing, QualityControl

Author Ryan Putney [cre, aut], Steven Eschrich [aut], Anders Berglund [aut]

Maintainer Ryan Putney <ryanputney@gmail.com>

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

git_branch devel

git_last_commit 5e1978f

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-03-28
getBarcodes

Contents

getBarcodes ......................................................... 2
getMethyLumiSet .................................................... 3
getSNparams ......................................................... 4
panelPlots .......................................................... 5
preprocess .......................................................... 7
subsetProbes ....................................................... 8

Index 10

getBarcodes  Get barcodes from idat file names

Description

A convenience function for retrieving simple barcodes from idat file names.

Usage

getBarcodes(path = getwd(), recurse = FALSE)

Arguments

path The path or a character vector to the directory or directories in which to find the
 idat files.
recurse logical; should the function check subdirectories to derive barcodes from any
 found idat files. The default is FALSE.

Details

Barcodes will be generated by all found ids in path(s). The default path is the current working
directory.

Value

A character vector of barcodes.

Author(s)

Ryan Putney <ryanputney@gmail.com>

See Also

getMethyLumiSet
getMethyLumiSet

Examples

```r
if(require(minfiData)){
  path <- system.file("extdata/5723646052", package="minfiData")
  barcodes <- getBarcodes(path = path)
}
```

getMethyLumiSet

Read idat files and create a MethyLumiSet object

Description

This a wrapper function for methylumIDAT that does not require a vector of barcodes to be provided.

Usage

```
getMethyLumiSet(path = getwd(), barcodes = NULL,
                 norm = c("none", "illumina", "SWAN", "dasen"),
                 bg.corr = TRUE)
```

Arguments

- `path`: The path to the directory containing the idat files.
- `barcodes`: A vector of barcodes specifying which idat’s to read.
- `norm`: Should normalization be done on the resulting MethyLumiSet. The default is "none".
- `bg.corr`: logical; if TRUE, an Illumina style background subtraction will be performed only if `norm` is set to 'illumina'. Otherwise, it is ignored. If background subtraction without any normalization is desired, the preprocess method must be used.

Details

If only `path` is provided, all idat’s found in the given folder will be pulled. If only barcodes is given, corresponding idat’s will be pulled from the current working directory. Both `path` and barcodes may be passed for finer control. The default is to pull all idat’s found in the current working directory.

Value

A MethyLumiSet object

Note

One would probably not normally want to use the preprocess option at this stage. It is more likely that a MethyLumiSet of the raw data will be desired. Then the preprocess method may be used to normalize the raw data or use background subtraction only on the raw data. See the vignette for example workflow.
getSNparams

**Estimate parameters for finite mixture of Skew-Normal distributions**

**Description**

Utilizes `smsn.mix` from the `mixsmsn` package to find the parameters for a finite mixture of skew normal distributions to model the overall distribution of signal intensities for a subset of probes on the Illumina Infinium HumanMethylation450. The probes may be subset by type and methylated or unmethylated. It can also be specified whether the SNP(rs), imprinted(idmr), or ch probes should be included or filtered out prior to parameter estimation.

**Usage**

```r
getSNparams(MethyLumiSet, allele = c('M', 'U'),
             type = c('I-red', 'I-green', 'II'),
             snps = TRUE, idmr = TRUE, ch = FALSE)
```

**Arguments**

- `MethyLumiSet` A MethyLumiSet object
- `allele` Should parameter estimation be done on the methylated or unmethylated signal intensities
- `type` Use the signal intensities for which probe type
- `snps` logical; should the rs probes be included in the dataset. The default is TRUE
- `idmr` logical; should the probes of imprinted gene loci be included in the dataset. The default is TRUE
- `ch` logical; should the ch probes be included in the dataset. The default is FALSE
panelPlots

Value

A Skew.normal object as returned by ssmn.mix from the mixsmsn package with the means and modes of the components added.

Author(s)

Ryan Putney <ryanputney@gmail.com>

References


See Also

subsetProbes

Examples

if(require('wateRmelon')) {
  data(melon)
  mixes.raw.meth.II <- getSNparams(melon[,1], 'M', 'II')
}

panelPlots

Plot the distributions of the probe intensities and the components of the skew-normal mixture model

Description

Creates a panel of nine plots. Six of the plots represent the density of either the methylated intensity or the unmethylated intensity given by one of three subsets of the 485,577 total probes. These subsets include Type I-red, Type I-green, and Type II. The remaining three distributions give the density of the beta-values for these same three subsets. Each of the nine plots optionally displays the distributions of the "rs" SNP probes and the probes associated with imprinted genes(Pidsley,2013) as a series of `tick` marks located above the x-axis.

Usage

panelPlots(MethyLumiSet, typeIRedModels, typeIGreenModels, typeIIModels, plot = c("panel", "frames"), samp.num = NULL, frame.nums = 1:9, norm = "", idmr = TRUE, snps = TRUE)
Arguments

**MethyLumiSet**  
The MethyLumiSet object from which the mixture models were derived

**typeIRedModels**  
A list of the Type I-red mixture models listed in the following order: methylated models followed by unmethylated models

**typeIGreenModels**  
A list of the Type I-green mixture models listed in the following order: methylated models followed by unmethylated models

**typeIIModels**  
A list of the Type II mixture models listed in the following order: methylated models followed by unmethylated models

**plot**  
Should the output consist of panel plots—one panel per sample or a single panel if `samp.num` is specified; or should the function output separate plots corresponding to the frames, given by `frame.nums`, for a single sample. The default is "panel". If set to "frames", `samp.num` must be specified

**samp.num**  
If plotting for a single sample is desired, for which sample. The number given simply refers to the MethyLumiSet column that corresponds to the sample of interest

**frame.nums**  
If `plot` is set to "frame", then `frame.nums` is a vector that specifies which frames of the panel to plot. The default is to plot all nine frames. The frames are numbered from 1 to 9 in column-major order starting with the top left. For example, to plot the four corners, use `frame.nums=c(1,3,7,9)`

**norm**  
A character string which will be displayed as part of the main title for each plot. Useful in indicated which normalization method was used for the modeled and plotted data

**idmr**  
logical; should the intensities of the idmr probes be plotted as a series of tick-marks above the x-axis. The default is TRUE

**snps**  
logical; should the intensities of the rs probes be plotted as a series of tick-marks above the x-axis. The default is TRUE

Value

No return value. Only plots are generated.

Note

Please refer to the vignette for an example workflow.

Author(s)

Ryan Putney <ryanputney@gmail.com>

References

preprocess

Normalize a MethyLumiSet object using some popular choices

Description

This is a wrapper function that allows normalizing of a MethyLumiSet using either a BeadStudio approximation, SWAN, or dasen. If desired, background correction only may be performed on the raw data.

Usage

```r
preprocess(MethyLumiSet, norm = c("none", "illumina", "SWAN", "dasen"),
  bg.corr = TRUE)
```

Arguments

- **MethyLumiSet**: A MethyLumiSet object
- **norm**: The normalization method to be used
- **bg.corr**: If TRUE, background subtraction using negative controls is performed. Ignored unless norm equals 'illumina' or 'none'

Details

Both Illumina style normalization via controls and the background correct method are handled by methylumi. The SWAN and dasen normalization methods are both performed by watermelon.

Examples

```r
if(require('minfiData')) {
  path <- system.file("extdata/5723646052", package="minfiData")
  methylumiset.raw <- getMethyLumiSet(path = path)
  mixes.raw.meth.I.red <- getSNparams(methylumiset.raw, 'M', 'I-red')
  mixes.raw.meth.I.green <- getSNparams(methylumiset.raw, 'M', 'I-green')
  mixes.raw.meth.II <- getSNparams(methylumiset.raw, 'M', 'II')
  mixes.raw.unmeth.I.red <- getSNparams(methylumiset.raw, 'U', 'I-red')
  mixes.raw.unmeth.I.green <- getSNparams(methylumiset.raw, 'U', 'I-green')
  mixes.raw.unmeth.II <- getSNparams(methylumiset.raw, 'U', 'II')
  mixes.I.red <- list(mixes.raw.meth.I.red, mixes.raw.unmeth.I.red)
  mixes.I.green <- list(mixes.raw.meth.I.green, mixes.raw.unmeth.I.green)
  mixes.II <- list(mixes.raw.meth.II, mixes.raw.unmeth.II)
  panelPlots(methylumiset.raw, mixes.I.red, mixes.I.green, mixes.II)
}
```
subsetProbes

Conveniently subset probes by type and retrieve the methylated or unmethylated intensities

Value
A MethyLumiSet

Author(s)
Ryan Putney <ryanputney@gmail.com>

References

See Also
getMethyLumiSet

Examples
if(require('watermelon')) {
  data(melon)
  melon.dasen <- preprocess(melon, norm = 'dasen')
}

subsetProbes(object, allele = c("M", "U"),
              type = c("I-red", "I-green", "II"),
              cg = TRUE, snps = TRUE, idmr = TRUE, ch = FALSE)
subsetProbes

Arguments

object A MethyLumiSet or MethylSet object
allele Should methylated or unmethylated data for the probes be returned.
type May be "I-red", "I-green", or "II".
cg Logical; Should the returned dataset contain the CpG probes. The default is TRUE
snps Logical; Should the returned dataset contain the rs probes. The default is TRUE
idmr Logical; should the returned dataset include probes that interrogate imprinted gene sites as given by Pidsley et al.(2013). The default is TRUE
ch Logical; should the returned dataset include the non-CpG (ch) probes. The default if FALSE

Value

A matrix

Author(s)

Ryan Putney <ryanputney@gmail.com>

References


See Also

getSNparams

Examples

```r
if(require('wateRmelon')) {
  data(melon)
  melon.meth.II <- subsetProbes(melon, 'M', 'II')
}
```
Index

* IO
  getMethylumiSet, 3

* aplot
  panelPlots, 5

* manip
  preprocess, 7
  subsetProbes, 8

* models
  getSNparams, 4

* utilities
  getBarcodes, 2
  getBarcodes, 2, 4
  getMethylumiSet, 2, 3, 8
  getSNparams, 4, 7, 9

  panelPlots, 5
  preprocess, 4, 7

  subsetProbes, 5, 8