methylumi

April 20, 2011

`extractBarcodeAndPosition`

Extract the Barcode and Position Information from Sentrix ID

Description

The sentrix IDs from an illumina sentrix array contain positional information that might be useful. This function simply extracts that information from the ID itself.

Usage

```
extractBarcodeAndPosition(sentrixids)
```

Arguments

`sentrixids`	A character vector of sentrix IDs that look like: 1632405013_R001_C001

Value

A data.frame with three columns:

`sentrix`	numeric, the sentrix ID
`row`	numeric, the sentrix row
`column`	numeric, the sentrix column

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`methylumiR`

Examples

```
extractBarcodeAndPosition(c('12341234_R001_C001'))
```

MethyLumi-accessors *methylumi accessors*

Description

These functions serve as getters and setters for information in methylumi classes.

Usage

```
betas(object)
pvals(object)
methylated(object)
unmethylated(object)
getHistory(object)
QCdata(object)
```

Arguments

`object`	an object of class MethyLumi or a subclass

Details

See the methods definitions in `MethyLumiSet` and `MethyLumiQC` for details.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`normalizeMethyLumiSet`, `MethyLumiSet`, `MethyLumiQC`, `eSet`

getAssayDataNameSubstitutions *Return a data.frame of AssayData name substitutions.*

Description

The Illumina methylation platforms use two distinct platforms, the "goldengate" platform and the "infinium" platform. Each of these uses different file formats as well as different assay techologies. To make the downstream data handling more straightforward and uniform between the two different systems, a simple mapping from the column names in the output files from the Illumina software is used to convert things from Red/Green or Cy5/Cy3 to unmethylated/methylated. This function simply returns that mapping.

Usage

```
getAssayDataNameSubstitutions()
```

Details

A file in the extdata directory called "substitutions.txt" contains two columns. The function loads this file and uses the first column as a match against column names in the data file (with the "sample part" removed). If matched, the second column gives the replacement.

Value

A data.frame with two columns, regex and replacement.

Author(s)

Sean Davis <seandavi@gmail.com>

Examples

```
getAssayDataNameSubstitutions()
```

`MethyLumi-class` *The base class for storing Illumina Methylation data*

Description

This class inherits from `eSet` from the Biobase package and is used as a base class for the other two methylumi classes, `MethyLumiSet` and `MethyLumiQC`.

Objects from the Class

The `MethyLumi` class is a virtual class and is not meant to be instantiated. Instead, one should instantiate a `MethyLumiSet` or a `MethyLumiQC` object.

Slots

`assayData`: Object of class `"AssayData"`

`phenoData`: Object of class `"AnnotatedDataFrame"`

`featureData`: Object of class `"AnnotatedDataFrame"` that will hold the annotation columns from the Beadstudio output, if they are available.

`experimentData`: Object of class `"MIAME"`

`annotation`: Object of class `"character"`; note that this slot is not currently used, but may be used in the future to store the character name of the annotation package, if available.

`.__classVersion__`: Object of class `"Versions"`

Extends

Class `"eSet"`, directly. Class `"VersionedBiobase"`, by class "eSet", distance 2. Class `"Versioned"`, by class "eSet", distance 3.

Methods

- **pvals<-** `signature(object = "MethyLumi", value = "matrix")`: Set the assayData slot of the same name and stores the P-values from BeadStudio
- **pvals** `signature(object = "MethyLumi")`: Get the assayData slot of the same name
- **betas<-** `signature(object = "MethyLumi", value = "matrix")`: Set the assayData slot of the same name and represents the methylation values for the samples, analogous to exprs() in gene expression data.
- **betas** `signature(object = "MethyLumi")`: Get the assayData slot of the same name
- **methylated<-** `signature(object = "MethyLumi", value = "matrix")`: Set the assayData slot that represents the Methylated single-channel signal
- **methylated** `signature(object = "MethyLumi")`: Get the assayData slot that represents the Methylated single-channel signal
- **unmethylated<-** `signature(object = "MethyLumi", value = "matrix")`: Set the assayData slot that represents the Unmethylated single-channel signal
- **unmethylated** `signature(object = "MethyLumi")`: Get the assayData slot that represents the Unmethylated single-channel signal
- **controlTypes** `signature(object = "MethyLumi"`: Find the unique control type beeds in the QCdata slot.
- **qcplot** `signature(object = "MethyLumi",what,...)`: Plot of QC data. This plot can be useful for diagnosing the problems associated with specific samples or arrays. The value for "what" is one of the control types (which can be found by using `controlTypes()` on the object.
- **summary** `signature(object = "MethyLumi",...)`: summary method for MethyLumi objects.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`methylumiR`, `MethyLumiSet`, `MethyLumiQC`, `eSet`

Examples

```
## The class structure
showClass("MethyLumi")
## read in some data
## Read in sample information
samps <- read.table(system.file("extdata/samples.txt",
                                package = "methylumi"),sep="\t",header=TRUE)
## Perform the actual data reading
## This is an example of reading data from a
## Sentrix Array format file (actually two files,
## one for data and one for QC probes)
mldat <- methylumiR(system.file('extdata/exampledata.samples.txt',
        package='methylumi'),
      qcfile=system.file('extdata/exampledata.controls.txt',
        package="methylumi"),
      sampleDescriptions=samps)
mldat
```

```
## Get history information
getHistory(mldat)
## Get QC data, which is another eSet-derived object
QCdata(mldat)
```

methylumi-package *Handle Illumina methylation data*

Description

This package contains a class structure for handling methylation data from Illumina as well as utility functions for loading the data from files generated by Illumina. Normalization that attempts to correct for dye bias is also included.

Important data classes include: `MethyLumiSet` and `MethyLumiQC`, both of which are subsets of the `MethyLumi` class, which is a subset of the `eSet` class.

A worked example of the use of the package can be found by typing: `openVignette()`.

A full listing of the available documentation can be obtained by typing `help.start()` and selecting `methylumi` from the `Packages` link or by typing `library(help="methylumi")`.

Details

Package:	methylumi
Type:	Package
License:	GPL

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

References

`http://watson.nci.nih.gov/~sdavis/software/R`

See Also

`Biobase`

MethyLumiQC-class *Class to hold Illumina Methylation QC data*

Description

This class inherits from the MethyLumi class (and therefore, from eSet in Biobase) and is designed to hold QC data from Illumina Beadstudio output. These data can be potentially useful when determining the cause for quality problems.

Objects from the Class

Objects can be created by calls of the form `new("MethyLumiQC", assayData, phenoData, featureData, experimentData, annotation, betas)`.

Slots

`assayData`: Object of class `"AssayData"`

`phenoData`: Object of class `"AnnotatedDataFrame"`

`featureData`: Object of class `"AnnotatedDataFrame"` containing the annotation columns from the Illumina Beadstudio output. In particular, the names of the probes describe the types of control probes.

`experimentData`: Object of class `"MIAME"`

`annotation`: Object of class `"character"`, not currently used

`.__classVersion__`: Object of class `"Versions"`

Extends

Class `"MethyLumi"`, directly. Class `"QCDataOrNULL"`, directly. Class `"eSet"`, by class "MethyLumi", distance 2. Class `"VersionedBiobase"`, by class "MethyLumi", distance 3. Class `"Versioned"`, by class "MethyLumi", distance 4.

Methods

qcplot `signature(object = "MethyLumiQC", what, ...)`: QC plots of various controltypes

controlTypes `signature(object = "MethyLumiQC")`: determine the character vector of control types from the QCdata information

initialize `signature(.Object = "MethyLumiQC")`

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`methylumiR`, `MethyLumiSet`, `MethyLumiQC`, `eSet`

Examples

```
showClass("MethyLumiQC")
```

`methylumiR` *Load data from Illumina methylation platform*

Description

This function is useful for loading Illumina methylation data into a MethyLumiSet object. Sample information can be supplied and will then be incorporated into the resulting phenoData slot.

Usage

```
methylumiR(filename,qcfile=NULL,sampleDescriptions = NULL,...)
```

Arguments

`filename` A filename of the excel-like file from BeadStudio

`qcfile` A filename of the excel-like file from BeadStudio

`sampleDescriptions`
A data.frame that contains at least one column, SampleID (case insensitive). This column MUST match the part of the column headers before the `.Avg_Beta`, etc. Also, if a column called SampleLabel (case insensitive), it is used for sample labels, IF the sampleLabel column contains unique identifiers

`...` Passed into `read.delim()`

Details

Used to construct a MethyLumiSet object....

Value

A MethyLumiSet object

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`MethyLumiSet-class`, `MethyLumiQC-class`

Examples

```
## Read in sample information
samps <- read.table(system.file("extdata/samples.txt",
                                package = "methylumi"),sep="\t",header=TRUE)
## Perform the actual data reading
## This is an example of reading data from an
## Sentrix Array format file (actually two files,
## one for data and one for QC probes)
mldat <- methylumiR(system.file('extdata/exampledata.samples.txt',package='methylumi'),
                    qcfile=system.file('extdata/exampledata.controls.txt',package="methyl
                    sampleDescriptions=samps)
mldat
```

`MethyLumiSet-class` *Class to hold Illumina Methylation data*

Description

This class inherits from the MethyLumi class (and therefore, from eSet in Biobase) and is designed to hold both the intensities and the calculated betas, as well as pvalues if present.

Objects from the Class

Objects can be created by calls of the form `new("MethyLumiSet", assayData, phenoData, featureData, experimentData, annotation, betas)`. An object of this type is the main storage class for methylation data from Illumina. Subsetting, etc., works as normal (rows represent genes, columns represent samples). There is also a rudimentary history tracking system that is modeled after that from the lumi package.

Slots

`QC`: Object of class `"QCDataOrNULL"`, containing either the `MethyLumiQC` object or NULL

`history`: Object of class `"data.frame"`, containing a running history of transforms to the data contained herein

`assayData`: Object of class `AssayData`

`phenoData`: Object of class `AnnotatedDataFrame`

`featureData`: Object of class `AnnotatedDataFrame`, containing the annotation columns from the Illumina Beadstudio output

`experimentData`: Object of class `MIAME`

`annotation`: Object of class `"character"`, not currently used

`.__classVersion__`: Object of class `"Versions"`

Extends

Class `"MethyLumi"`, directly. Class `"eSet"`, by class "MethyLumi", distance 2. Class `"VersionedBiobase"`, by class "MethyLumi", distance 3. Class `"Versioned"`, by class "MethyLumi", distance 4.

Methods

[`signature(x = "MethyLumiSet")`: subsetting, genes as rows, samples as columns

betas<- `signature(object = "MethyLumiSet", value = "matrix")`: Set the assayData slot of the same name

betas `signature(object = "MethyLumiSet")`: Get the assayData slot of the same name

boxplot `signature(x = "MethyLumiSet")`: boxplot of all sample betas

combine `signature(x = "MethyLumiSet", y = "MethyLumiSet")`

corplot `signature(x = "MethyLumiSet")`

exprs<- `signature(object = "MethyLumiSet", value = "matrix")`: alias for "<-betas"

exprs `signature(object = "MethyLumiSet")`: alias for "betas"

getHistory `signature(object = "MethyLumiSet")`: returns a data.frame containing the history for this object

hist `signature(x = "MethyLumiSet")`: histogram of the betas for the data

initialize `signature(.Object = "MethyLumiSet")`

pairs `signature(x = "MethyLumiSet")`: pairs plot of the betas for the object. Note that pairs plots of more than a few samples are not very useful.

plotSampleIntensities `signature(x = "MethyLumiSet")`: The intensities as output by the Beadstudio software often show a considerable amount of dye bias. This method shows a graphical example of this dye bias. In short, for each of the Cy3 and Cy5 channels, a cutoff in beta is used to calculate which Cy3 and Cy5 values should be plotted at high-methylation and low-methylation status. Any offset between Cy3 and Cy5 when plotted in this way likely represents dye bias and will lead to biases in the estimate of beta.

QCdata<- `signature(object = "MethyLumiSet", value = "MethyLumiQC")`: assign QC data to the QC slot

QCdata `signature(object = "MethyLumiSet")`: retrieve the QC data.

show `signature(object = "MethyLumiSet")`

methylated<- `signature(object = "MethyLumiSet", value = "matrix")`: Set the assayData slot associated with methylated intensity

methylated `signature(object = "MethyLumiSet")`: Get the assayData slot associated with methylated intensity

unmethylated<- `signature(object = "MethyLumiSet", value = "matrix")`: Set the assayData slot associated with unmethylated intensity

unmethylated `signature(object = "MethyLumiSet")`: Get the assayData slot associated with unmethylated intensity

qcplot `signature(object = "MethyLumiSet", what, ...)`: QC plots of various controltypes

controlTypes `signature(object = "MethyLumiSet")`: determine the character vector of control types from the QCdata information

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

`methylumiR`, `normalizeMethyLumiSet`, `MethyLumiSet`, `MethyLumiQC`, `eSet`

Examples

```
showClass("MethyLumiSet")
```

mldat *Example SAM format Illumina methylation dataset*

Description

This is an example MethyLumiSet object.

Usage

```
data(mldat)
```

Examples

```
data(mldat)
```

normalizeMethyLumiSet *Normalize a MethyLumiSet, accounting for dye bias*

Description

The Illumina GoldenGate methylation platform uses two colors, one to represent the unmethylated state and the other to represent the methylated state. This function corrects that dye bias and recalculates the betas based on the corrected intensities. It is probably not optimal for Infinium data since the methylated and unmethylated signals are in the same channel for that platform.

Usage

```
normalizeMethyLumiSet(x, beta.cuts = c(0.2, 0.8), mapfun = c("atan", "ratio"))
```

Arguments

x	A MethyLumiSet object
beta.cuts	Two numeric values with the first less than the second and between 0 and 1, representing the beta cutoffs that will be used when determining the median intensities to which to correct. See details below.
mapfun	Either "atan" or "ratio". See details below.

Details

The Illumina GoldenGate methylation platform uses two colors, one to represent the unmethylated state and the other to represent the methylated state. This function corrects that dye bias and recalculates the betas based on the corrected intensities.

As a first step, the medians for each of Cy3 and Cy5 are calculated at high and low betas, representing the (nearly) fully methylated state and the (nearly) fully unmethylated states. Values of Cy3 and Cy5 that are negative are set to zero for this process. Then, the Cy5 medians are adjusted to match those of the Cy3 channel, thereby correcting the dye bias.

To map the new intensities back to betas, one of two map functions can be used. The default is the atan(Cy3/Cy5). The ratio maps using the function (Cy3/Cy3+Cy5). The differences should be very small, but we feel that the atan map function is probably the mathematically appropriate way of doing this.

Note that this normalization method is not optimal for Infinium methylation data and should probably not be used for that platform.

Value

A new "MethyLumiSet" that contains the corrected betas and the adjusted intensities.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

Examples

```
## Read in sample information
samps <- read.table(system.file("extdata/samples.txt",
                                package = "methylumi"),sep="\t",header=TRUE)
## Perform the actual data reading
## This is an example of reading data from an
## Sentrix Array format file (actually two files,
## one for data and one for QC probes)
mldat <- methylumiR(system.file('extdata/exampledata.samples.txt',package='methylumi'),
                    qcfile=system.file('extdata/exampledata.controls.txt',package="methyl
                    sampleDescriptions=samps)
mldatnorm <- normalizeMethyLumiSet(mldat)
```

plotSampleIntensities

Plot the sample intensities.

Description

The Illumina methylation platforms all show a significant dye bias. The plotSampleIntensities method shows the density plots for the two channels allowing direct visualization of the effect.

Usage

```
plotSampleIntensities(x,beta.cuts,s)
```

Arguments

`x`	an object of class MethyLumi or a subclass
`beta.cuts`	cutoffs for low and high beta values
`s`	sample number to plot

Examples

```
data(mldat)
plotSampleIntensities(mldat,s=1)
```

qcplot *Methods for dealing with control data for Illumina methylation data.*

Description

The qcplot function simply generates a plot of the control probe information for a given controlType.

Usage

```
qcplot(object,controltype,...)
controlTypes(object,...)
```

Arguments

object	An object of class MethyLumiSet or MethyLumiQC
controltype	A single character value representing the bead type to plot from the quality control data. The available types are accessible via the controlTypes method.
...	passed to plot function

Details

The descriptions of the various control types can be obtained from the Illumina methylation user's guides.

Author(s)

Sean Davis <sdavis2@mail.nih.gov>

See Also

MethyLumiSet, MethyLumiQC

Examples

```
data(mldat)
controlTypes(mldat)
qcplot(mldat,controlTypes(mldat)[3])
```

Index