funtooNorm Package

March 22, 2016

`funtooNorm` *funtooNorm*

Description

The funtooNorm Package provides a normalization method for data arising from the Illumina Infinium Human Methylation 450 BeadChip (Illumina 450K), including explicit considerations of differences between tissues or cell types. This method should only be used when the data set contains samples fom multiple different tissues or cell types.

Details

Package:	funtooNorm
Type:	Package
Version:	0.99.4
Date:	2016-03-28
License:	GPL-3

`SampleSet-class` *SampleSet are S3 objects define for the purpose of runing funtooNorm algorithm, they are list containing signal data and different variables usefull for funtooNorm. The data is separated into the 3 probes type, having 2 channels (methylated and unmethylated ie : A and B. We then define then the 6 (2*3) labels: AIGrn BIGrn AIRed BIRed AII BII*

Description

SampleSet are S3 objects define for the purpose of runing funtooNorm algorithm, they are list containing signal data and different variables usefull for funtooNorm. The data is separated into the 3 probes type, having 2 channels (methylated and unmethylated ie : A and B. We then define then the 6 (2*3) labels: AIGrn BIGrn AIRed BIRed AII BII

Value

a SampleSet object

Slots

`type` character: is 'minfi' or 'GenomeStudio'

`sampleNames` character vector: contain the list of sample names in order used

`sampleSize` numeric: the number of samples

`nPos` numeric: the number of positions in the ILLUMINA chip

`cell_type` list: list matching each sample to define the categories

`qntllist` numeric: vector of ordered quantiles

`quantiles` numeric: list of 6 auqntiles tables for 6 type of signals

`ctl.covmat` numeric: covariance matrix for the model fit

`signal` numeric: list of 6 signal tables the 6 type of signals

Examples

```
showClass("SampleSet")
```

`agreement` *Function to measure intra-replicate agreement in methylation data.*

Description

Function to measure intra-replicate agreement in methylation data.

Usage

```
agreement(Beta, individualID)
```

Arguments

`individualID`	: a vector where 2 replicates have the exact same value for two technical replicate. Order of samples should nmatch the samples (columns) in Beta
`Matrix`	with beta-values, rows corresponding to probes, columns corresponding to samples.

Details

We expect that the values returned by the agreement function after normalization by funtooNorm to be smaller than before.

Value

The average value of the square distance between replicates: a measure of agreement between replicates in methylation data.

fromGenStudFiles *create a sample set from GenomeStudio files*

Description

create a sample set from GenomeStudio files

Usage

```
fromGenStudFiles(controlProbeFile, signalFile, cell_type)
```

Arguments

controlProbeFile	file exported from GenomeStudio
signalFile	file exported from GenomeStudio with the exact same samples
cell_type	this vector should have names matching all the sample in the files from genome studios, and at least 2 different cell types.

Value

a SampleSet object

Examples

```
 myNewSampleSet <- fromRGChannelSet("ControlProbeProfile.txt",
"SignalIntensity.txt",cell_type)
```

fromRGChannelSet *create a SampleSet from RGSet from minfi package*

Description

create a SampleSet from RGSet from minfi package

Usage

```
fromRGChannelSet(myRGChannelSet)
```

Arguments

myRGChannelSet,	from mini package, should contain a cell_type vector in it s phenotypes data pData

Value

a SampleSet object

Examples

```
myNewSampleSet <- fromRGChannelSet(objectOfTypeRGChannelSet)
```

`funtooNorm` *This function apply the normalization method central to the package on each signal*

Description

This function apply the normalization method central to the package on each signal

Usage

```
funtooNorm(object, type.fits = "PCR", ncmp = 4, force = FALSE)
```

Arguments

`object`	of type SampleSet
`type.fits`	can be "PCR" or "PLS" (default "PCR")
`ncmp`	number of component use for anaysis (default 4)
`force`	set it to TRUE in order to re-compute the normalisation whent it is already done

Value

a SampleSet containing the normalised signal

Examples

```
mySampleSet  <- funtooNorm(mySampleSet)
```

`getLogSigA` *internal function to get the signal A*

Description

internal function to get the signal A

Usage

```
getLogSigA(signal)
```

Arguments

`signal`

Value

all signal A in the good order

`getLogSigB` *internal function to get the signal B*

Description

internal function to get the signal B

Usage

```
getLogSigB(signal)
```

Arguments

```
signal
```

Value

all signal B in the good order

`getNormBeta` *compute the beta value after normalization for each position and each sample*

Description

compute the beta value after normalization for each position and each sample

Usage

```
getNormBeta(object, offset = 100)
```

Arguments

`object`	of type SampleSet
`offset`	default is 100 as Illumina standard

Value

a matrix containing beta after normalization value for each CpG position and each samples

Examples

```
myNormBetaMatrix  <-getNormBeta(mySampleSet)
```

`getNormM` *compute the M value after normalization for each position and each sample*

Description

compute the M value after normalization for each position and each sample

compute the beta value after normalization for each position and each sample

Usage

```
getNormM(object, offset = 100)

getNormM(object, offset = 100)
```

Arguments

`object`	of type SampleSet
`offset`	default is 100 as Illumina standard
`object`	of type SampleSet
`offset`	default is 100 as Illumina standard

Value

a matrix containing M after normalization value for each position and each samples log(Meth/Unmeth)

a matrix containing M after normalization value for each position and each samples

---

`getRawBeta` *compute the beta value of the raw signal for each position and each sample*

Description

compute the beta value of the raw signal for each position and each sample

Usage

```
getRawBeta(object, offset = 100)
```

Arguments

`object`	of type SampleSet
`offset`	default is 100 as Illumina standard

Value

a matrix containing beta value for each position and each samples

Examples

```
myRawBetaMatrix  <- getRawBeta(mySampleSet)
```

`getSampleSetAnnotation` *Get the SampleSet Annotation in the same order as the beta matrix*

Description

Get the SampleSet Annotation in the same order as the beta matrix

Usage

```
getSampleSetAnnotation()
```

Arguments

`object`	of type SampleSet

Value

annotation

Examples

```
getSampleSetAnnotation(mySampleSet)
```

`getSnpM` *compute the M value after normalization for each SNP position and each sample*

Description

compute the M value after normalization for each SNP position and each sample

Usage

```
getSnpM(object)
```

Arguments

`object`	of type SampleSet
`offset`	default is 100 as Illumina standard

Value

a matrix containing M after normalization value for each SNP of the chip and each sample log(Meth/Unmeth)

`plotValidationGraph` *Plot a series of graphs with different numbers of components for each signal*

Description

Plot a series of graphs with different numbers of components for each signal

Usage

```
plotValidationGraph(object, type.fits = "PCR", file = "")
```

Arguments

`object`	of type SampleSet
`type.fits`	can be "PCR" or "PLS" (default "PCR")
`file`	if not empty will writte a pdf at this place

Examples

```
plotValidationGraph(mySampleSet,file="myPlots.pdf")
```

`print.SampleSet` *Print information about the SampleSet*

Description

Print information about the SampleSet

Usage

```
## S3 method for class 'SampleSet'
print(object)
```

Arguments

`object`	of type SampleSet

Examples

```
mySampleSet
```

Index