funtooNorm Package

April 4, 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

Author(s)

Celia Greenwood, Raphael Poujol, Stepan Grinek, Kathleen Oros Klein

`SampleSet-class`	*SampleSet is an S3 class defined for the purpose of running the funtooNorm algorithm. They are lists containing signal data and different variables useful for funtooNorm. The data is separated into the 3 probes types, each 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 is an S3 class defined for the purpose of running the funtooNorm algorithm. They are lists containing signal data and different variables useful for funtooNorm. The data is separated into the 3 probes types, each 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

`annotation` IlluminaMethylationAnnotation: the annotation object from mnfi package

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

`qntllist` numeric: vector of ordered quantiles

`quantiles` numeric: list of 6 quantiles 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

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

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.

Examples

```
agreement(cbind(rnorm(n = 10),rnorm(n = 10),rnorm(n = 10)),c(1,1,1))
```

---

fromGenStudFiles *create a SampleSet from GenomeStudio files*

Description

create a SampleSet from GenomeStudio files

Usage

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

Arguments

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

Value

a SampleSet object

`fromRGChannelSet` *create a SampleSet from RGChannelSet from minfi package*

Description

create a SampleSet from RGChannelSet from minfi package

Usage

```
fromRGChannelSet(myRGChannelSet)
```

Arguments

`myRGChannelSet`
: RGChannelSet, from minfi package, should contain a cell_type vector in it s phenotypes data pData

Value

a SampleSet object

Examples

```
require(funtooNorm)
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
```

`funtooNorm` *This function applies the normalization method central to the package to each signal. The chrY have a deserve a specific treatment, men are asses using the median beta estimation on the raw data positions with a cutoff at 60 percent. We perform on the mens a quantile normalization and we do not change the women values*

Description

This function applies the normalization method central to the package to each signal. The chrY have a deserve a specific treatment, men are asses using the median beta estimation on the raw data positions with a cutoff at 60 percent. We perform on the mens a quantile normalization and we do not change the women values

Usage

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

Arguments

object	of type SampleSet
type.fits	can be "PCR" or "PLS" (default "PCR")
ncmp	number of components used in the analysis (default 4)
force	set it to TRUE in order to re-compute the normalisation whent it is already done
sex	boolean vector: when not null force the chrY normalization to use treat the TRUE values as mens

Value

a SampleSet containing the normalised signal

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
mySampleSet=funtooNorm(mySampleSet)
```

---

getGRanges *Return a list*

Description

Return a list

Usage

```
getGRanges(object)
```

Arguments

```
object
```

Value

a GRange object of all the methylated positions

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
gr=getGRanges(mySampleSet)
```

`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

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
b=getNormBeta(funtooNorm(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

Usage

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

Arguments

`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 log2(Meth/Unmeth)

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
m=getNormM(funtooNorm(mySampleSet))
```

`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 the raw beta value for each position and each samples

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
r=getRawBeta(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

Value

a matrix containing M after normalization value for each SNP of the chip and each sample log2(Meth/Unmeth)

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
snp=getSnpM(funtooNorm(mySampleSet))
```

---

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 write a pdf using this name, path can be included

Examples

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
plotValidationGraph(mySampleSet)
```

---

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

```
require(minfiData)
pData(RGsetEx)$cell_type <- rep(c("type1","type2"),3)
mySampleSet=fromRGChannelSet(RGsetEx)
mySampleSet
```

Index