crlmm

April 19, 2010

`ABset-class` *Class "ABset"*

Description

Containter for quantile-normalized intensities

Objects from the Class

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

Slots

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

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

`featureData`: Object of class `"AnnotatedDataFrame"`

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

`annotation`: Object of class `"character"`

`protocolData`: Object of class `"AnnotatedDataFrame"`

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

Extends

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

Methods

A `signature(object="ABset")`: accessor for the quantile-normalized intensities of allele A for polymorphic probes and the quantile normalized intensities for the copy number probes.

"A<-" `signature(object="ABset", value="matrix")`: replacement method for the A allele intensities.

"B<-" `signature(object="ABset", value="matrix")`: replacement method for the B allele intensities.

B `signature(object="ABset")`: accessor for the quantile-normalized intensities of allele B for polymorphic probes.

plot `signature(x = "ABset", y = "CopyNumberSet")`: ...

Author(s)

R. Scharpf

Examples

```
showClass("ABset")
```

---

batch *Stores information on the batch (e.g., chemistry plate) used to process samples.*

Description

Parameters for estimating copy number are locus- and batch-specific.

Usage

```
batch(object)
```

Arguments

object	An object of class `CrlmmSetList` or `CopyNumberSet`.

Value

Typically, a character string or factor.

Author(s)

Rob Scharpf

---

celDates *Extract dates from the cel file header*

Description

Extract dates from the cel file header.

Usage

```
celDates(celfiles)
```

Arguments

celfiles	CEL file names. Must specify the complete path.

Value

date-time class POSIXt

Author(s)

R. Scharpf

See Also

read.celfile.header, *POSIXt*

`cnrma` *Quantile normalizes the intensities for the nonpolymorphic probe to a HapMap reference distribution.*

Description

Quantile normalizes the intensities for the nonpolymorphic probe to a HapMap reference distribution.

Usage

```
cnrma(filenames, cdfName, sns, seed = 1, verbose=FALSE)
```

Arguments

`filenames`	filenames with complete path
`cdfName`	Only 'genomewidesnp6' allowed
`sns`	the sample names. If missing, the basename of the filenames is used.
`seed`	seed for sampling the intensities to calculate skewness
`verbose`	logical

Value

A list. First element is the matrix of quantile-normalized intensities. The second element is the skew.

Note

Not tested

Author(s)

Rob Scharpf

Examples

```
library(genomewidesnp6Crlmm)
library(hapmapsnp6)
path <- system.file("celFiles", package="hapmapsnp6")
celFiles <- list.celfiles(path, full.names=TRUE)
cnrmaResult <- cnrma(celFiles, cdfName="genomewidesnp6")
```

`computeCopynumber` *Computes copy number*

Description

A function that transforms the quantile-normalized fluorescence intensities of the polymorphic and nonpolymorphic probes to a copy number scale.

Usage

```
computeCopynumber(object, CHR, bias.adj=FALSE, batch, SNRmin=5, cdfName,
...)
```

Arguments

`object`	object of class `CrlmmSetList`.
`CHR`	Chromosome (an integer). Use 23 for X and 24 for Y.
`bias.adj`	Logical. When TRUE, an extra iteration is performed (see details) .
`batch`	Factor. Must be the same length as the number of samples in the object.
`SNRmin`	The minimum value for the SNR – we suggest 5. Samples with SNR below SNRmin are dropped.
`cdfName`	Annotation package
`...`	arguments to `.computeCopynumber`.

Details

One may altenatively use the `update` method (a wrapper for `computeCopynumber`) to add a `CopyNumberSet` object to the `CrlmmSetList` object.

This function requires 10 or more samples to estimate model parameters. Preferably, 70+ samples would be processed together in a batch.

This function translates the quantile-normalized fluorescence intensities to the scale of copy number. We assume that for any given locus the median copy number is two for each batch. When `bias.adj=TRUE`, an extra iteration is performed whereby samples with a high posterior probability of having a non-normal copy number are excluded. The resulting within-genotype estimators of location and scale are more robust to a large number of samples having a copy number variant.

For details, see the technical report:

Scharpf RB, Ruczinski I, Carvalho B, Doan B, Chakravarti A, and Irizarry R. A multilevel model to address batch effects in copy number estimation using SNP arrays (http://www.bepress.com/cgi/viewcontent.cgi?article=

Value

An object of class `CrlmmSetList`.

Author(s)

Rob Scharpf

See Also

`update`, `CrlmmSetList`, `CopyNumberSet`, `.computeCopynumber`

CopyNumberSet-class
Class "CopyNumberSet"

Description

Container for allele-specific estimates of copy number and the corresponding uncertainty

Objects from the Class

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

Slots

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

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

`featureData`: Object of class `"AnnotatedDataFrame"`

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

`annotation`: Object of class `"character"`

`protocolData`: Object of class `"AnnotatedDataFrame"`

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

Extends

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

Methods

batch `signature(object="CopyNumberSet")`: Extracts the batch information used to estimate copy number.

CA `signature(object = "CopyNumberSet", ...)`: Extracts the copy number for allele 'A' at polymorphic loci. At nonpolymorphic loci, the total copy number is returned.

CA<- `signature(object = "CopyNumberSet", value="matrix")`: Replaces CA matrix with supplied value.

CB `signature(object = "CopyNumberSet", ...)`: Extracts copy number for allele 'B' at polymorphic loci. NAs are returned for nonpolymorphic loci.

CB<- `signature(object = "CopyNumberSet", value="matrix")`: Replaces CB matrix with supplied value .

chromosome `signature(object = "CopyNumberSet")`: Extract chromosome.

copyNumber `signature(object = "CopyNumberSet")`: Returns CA + CB.

ellipse `signature(x = "CopyNumberSet", ...)`: Extracts parameters from `featureData` slot and draws prediction regions for the supplied copy number.

plot `signature(x = "ABset", y = "CopyNumberSet")`: Physical position

position `signature(object = "CopyNumberSet")`: physical position

Author(s)

R. Scharpf

Examples

```
showClass("CopyNumberSet")

## Not run:
##returns the copy number for allele A at polymorphic loci
CA(object[snpIndex(object), ])
##returns the total copy number at nonpolymorphic loci
CA(object[cnIndex(object), ])

## End(Not run)
```

`crlmmIllumina` *Genotype Illumina Infinium II BeadChip data with CRLMM*

Description

This implementation of the CRLMM is especially designed for data from Illumina Infinium II BeadChips.

Usage

```
crlmmIllumina(RG, XY, stripNorm=TRUE, useTarget=TRUE,
      row.names=TRUE, col.names=TRUE,
      probs=c(1/3, 1/3, 1/3), DF=6, SNRMin=5,
      gender=NULL, seed=1, save.it=FALSE, load.it=FALSE,
      intensityFile, mixtureSampleSize=10^5,
      eps=0.1, verbose=TRUE, cdfName, sns, recallMin=10,
      recallRegMin=1000, returnParams=FALSE, badSNP=0.7)
```

Arguments

`RG`	`NChannelSet` containing R and G bead intensities
`XY`	`NChannelSet` containing X and Y bead intensities
`stripNorm`	'logical'. Should the data be strip-level normalized?
`useTarget`	'logical' (only used when `stripNorm=TRUE`). Should the reference HapMap intensities be used in strip-level normalization?
`row.names`	'logical'. Use rownames - SNP names?
`col.names`	'logical'. Use colnames - Sample names?
`probs`	'numeric' vector with priors for AA, AB and BB.
`DF`	'integer' with number of degrees of freedom to use with t-distribution.
`SNRMin`	'numeric' scalar defining the minimum SNR used to filter out samples.
`gender`	'integer' vector, with same length as 'filenames', defining sex. (1 - male; 2 - female)

seed	'integer' scalar for random number generator (used to sample mixtureSampleSize SNPs for mixture model.
save.it	'logical'. Save preprocessed data?
load.it	'logical'. Load preprocessed data to speed up analysis?
intensityFile	'character' with filename of preprocessed data to be saved/loaded.
mixtureSampleSize	'integer'. The number of SNP's to be used when fitting the mixture model.
eps	Minimum change for mixture model.
verbose	'logical'.
cdfName	'character' defining the chip annotation (manifest) to use ('human370v1c', human550v3b', 'human650v3a', 'human1mv1c', 'human370quadv3c', 'human610quadv1b', 'human660quadv1a' 'human1mduov3b')
sns	'character' vector with sample names to be used.
recallMin	'integer'. Minimum number of samples for recalibration.
recallRegMin	'integer'. Minimum number of SNP's for regression.
returnParams	'logical'. Return recalibrated parameters.
badSNP	'numeric'. Threshold to flag as bad SNP (affects batchQC)

Details

Note: The user should specify either the RG or XY intensities, not both. Alternatively if crlmmIllumina has been run already with save.it=TRUE, the preprocessed data can be loaded from file by specifying load.it=TRUE and intensityFile (RG or XY are not needed in this case).

Value

A SnpSet object which contains

calls	Genotype calls (1 - AA, 2 - AB, 3 - BB)
callProbability	confidence scores 'round(-1000*log2(1-p))'

in the assayData slot and

SNPQC	SNP Quality Scores
batchQC	Batch Quality Scores

along with center and scale parameters when returnParams=TRUE in the featureData slot.

Author(s)

Matt Ritchie

References

Carvalho B, Bengtsson H, Speed TP, Irizarry RA. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. Biostatistics. 2007 Apr;8(2):485-99. Epub 2006 Dec 22. PMID: 17189563.

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

Examples

```
## crlmmOut = crlmmIllumina(RG)
```

`crlmm-package` *Genotype Calling via CRLMM Algorithm*

Description

Faster implementation of CRLMM specific to SNP 5.0 and 6.0 arrays.

Details

Index:

```
crlmm-package           New implementation of the CRLMM Algorithm.
crlmm                   Genotype SNP 5.0 or 6.0 samples.
calls                   Accessor for genotype calls.
confs                   Accessor for confidences.
```

The 'crlmm' package reimplements the CRLMM algorithm present in the 'oligo' package. This implementation primes for efficient genotyping of samples on SNP 5.0 and SNP 6.0 Affymetrix arrays.

To use this package, the user must have additional data packages: 'genomewidesnp5Crlmm' - SNP 5.0 arrays 'genomewidesnp6Crlmm' - SNP 6.0 arrays

Author(s)

Rafael A Irizarry Maintainer: Benilton S Carvalho <bcarvalh@jhsph.edu>

References

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

`crlmm` *Genotype oligonucleotide arrays with CRLMM*

Description

This is a faster and more efficient implementation of the CRLMM algorithm, especially designed for Affymetrix SNP 5 and 6 arrays (to be soon extended to other platforms).

Usage

```
crlmm(filenames, row.names=TRUE, col.names=TRUE,
      probs=c(1/3, 1/3, 1/3), DF=6, SNRMin=5,
      gender=NULL, save.it=FALSE, load.it=FALSE,
      intensityFile, mixtureSampleSize=10^5,
      eps=0.1, verbose=TRUE, cdfName, sns, recallMin=10,
      recallRegMin=1000, returnParams=FALSE, badSNP=0.7)
```

Arguments

filenames	'character' vector with CEL files to be genotyped.
row.names	'logical'. Use rownames - SNP names?
col.names	'logical'. Use colnames - Sample names?
probs	'numeric' vector with priors for AA, AB and BB.
DF	'integer' with number of degrees of freedom to use with t-distribution.
SNRMin	'numeric' scalar defining the minimum SNR used to filter out samples.
gender	'integer' vector, with same length as 'filenames', defining sex. (1 - male; 2 - female)
save.it	'logical'. Save preprocessed data?
load.it	'logical'. Load preprocessed data to speed up analysis?
intensityFile	'character' with filename to be saved/loaded - preprocessed data.
mixtureSampleSize	Number of SNP's to be used with the mixture model.
eps	Minimum change for mixture model.
verbose	'logical'.
cdfName	'character' defining the CDF name to use ('GenomeWideSnp5', 'GenomeWideSnp6')
sns	'character' vector with sample names to be used.
recallMin	Minimum number of samples for recalibration.
recallRegMin	Minimum number of SNP's for regression.
returnParams	'logical'. Return recalibrated parameters.
badSNP	'numeric'. Threshold to flag as bad SNP (affects batchQC)

Value

A SnpSet object.

calls	Genotype calls (1 - AA, 2 - AB, 3 - BB)
confs	Confidence scores 'round(-1000*log2(1-p))'
SNPQC	SNP Quality Scores
batchQC	Batch Quality Score
params	Recalibrated parameters

References

Carvalho B, Bengtsson H, Speed TP, Irizarry RA. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. Biostatistics. 2007 Apr;8(2):485-99. Epub 2006 Dec 22. PMID: 17189563.

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

Examples

```
## this can be slow
if (require(genomewidesnp5Crlmm) & require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
  cels <- list.celfiles(path, full.names=TRUE)
  (crlmmOutput <- crlmm(cels))
}
```

CrlmmSetList-class *Class "CrlmmSetList"*

Description

Container for quantile normalized intensities and genotype calls.

Objects from the Class

Objects from the class are created by calls to the `crlmmWrapper` function.

Slots

`.Data`**:** Object of class `"list"` ~~

Details

Instances of `CrlmmSetList` are a list with two elements:

1. an object of class `ABset`

2. an object of class `SnpSet`

The `featureNames` and `sampleNames` of both objects are required to be identical.

Quantile-normalized intensities for the copy number probes are stored in the assay data element A of the ABset object. Corresponding rows for the B allele are recorded as NAs. Genotype calls for copy number probes in the SnpSet object are recorded as NAs.

Extends

Class `"list"`, from data part. Class `"vector"`, by class "list", distance 2. Class `assayData`, by class "list", distance 2.

Methods

"[" `signature(x = "CrlmmSetList")`: subsets both the features and samples of each element in th elist

"$" `signature(x = "CrlmmSetList")`: used to access element from the `featureData` of the `CopyNumberSet` element. In particular, useful for extract the SNP- and batch-specific parameters used to estimate copy number that are currently stored in the `featureData` of `CopyNumberSet` objects.

"addFeatureAnnotation" `signature(object = "CrlmmSetList")`: Creates an object of class AnnotatedDataFrame from the CrlmmSetList object. The new annotation object contains information on chromosome and physical position and has the same ordering of rows as the CrlmmSetList object.

"A" `signature(object = "CrlmmSetList")`: extracts the quantile normalized intensities for the A allele in the `ABset` element.

"A<-" `signature(object = "CrlmmSetList", value = "matrix")`: Replacement method for the A intensities.

"B" `signature(object = "CrlmmSetList")`: extracts the quantile normalized intensities for the B allele in the `ABset` element.

"B<-" `signature(object = "CrlmmSetList", value = "matrix")`: Replacement method for the B intensities.

"batch" `signature(object = "CrlmmSetList")`: extracts the batch information used to estimate copy number.

"CA" `signature(object = "CrlmmSetList")`: extracts the copy number for allele A at polymorphic loci. For nonpolymorphic probes, CA returns the total copy number.

"CB" `signature(object = "CrlmmSetList")`: extracts the copy number for allele B at polymorphic loci. For nonpolymorphic probes, CB returns 'NA'.

"CA<-" `signature(object = "CrlmmSetList", value = "matrix")`: Replacement method for the CA estimates.

"CB<-" `signature(object = "CrlmmSetList", value = "matrix")`: Replacement method for the CB estimates.

"calls" `signature(object = "CrlmmSetList")`: extracts the genotype calls from the `SnpSet` element.

"cnIndex" `signature(object = "CrlmmSetList")`: returns the row indices of the copy number probes.

"combine" `signature(x = "CrlmmSetList", y = "CrlmmSetList")`: combines objects of the class.

"confs" `signature(x = "CrlmmSetList")`: confidence scores for crlmm genotype calls.

"nrow" `signature(x = "CrlmmSetList")`: Number of rows (features) of each element in the list.

"ncol" `signature(x = "CrlmmSetList")`: Number of columns (samples) of each element in the list.

"plot" `signature(x = "CrlmmSetList")`: For A versus B scatterplots.

"points" `signature(x = "CrlmmSetList")`: For A versus B scatterplots.

"sampleNames" `signature(object = "CrlmmSetList")`: Accessor for the column identifiers of the assay data elements.

"show" `signature(object = "CrlmmSetList")`: Shows the `ABset` and `SnpSet` elements of the list.

"snpIndex" `signature(object = "CrlmmSetList")`: returns the row indices of the polymorphic probes.

"update" `signature(object = "CrlmmSetList")`: This method calls `computeCopynumber` to compute copy number for the `crlmmSetList` object. The object returned by this method is itself an instance of class `CrlmmSetList`. The third element of the list is an instance of `CopyNumberSet` containing the locus-level, allele-specific estimates of copynumber.

"update" `signature(object = "character")`: This method loads the CrlmmSetList file given as the argument to object – the object is a character string or a vector of character strings that provides the complete path to the CrlmmSetList file. The CrlmmSetList object is loaded and updated with copy number estimates – an object of class CopyNumberSet becomes the third element in the CrlmmSetList object. After updating the CrlmmSetList object, the object is saved to the same path.

Warnings

`combine` will produce a warning as the `featureData` are not easily combined and therefore removed. This method needs improvement.

Author(s)

R. Scharpf

See Also

See also `"crlmmWrapper"`, `"ABset"`

Examples

```
showClass("CrlmmSetList")
```

`crlmmWrapper` *Wrapper to quantile normalize and genotype Affymetrix 6.0 cel files.*

Description

Quantile normalizes the cel files to a target reference distribution (both polymorphic and nonpolymorphic loci). Genotype calls and confidence scores are assigned using the crlmm algorithm.

Usage

```
crlmmWrapper(filenames, cdfName, load.it=FALSE,
save.it=FALSE, splitByChr=TRUE, crlmmFile, intensityFile, rgFile, ...)
```

Arguments

`filenames`	Complete path to cel files
`cdfName`	Annotation package (currently, only 'genomewidesnp6' is supported). See `crlmm:::validCdfNa`
`load.it`	Logical. If TRUE, the wrapper function will try to load the 'intensityFile' and 'crlmmFile'. When FALSE, the quantile normalization and genotype calling steps are recomputed regardless of whether these objects exist.
`save.it`	Logical. Whether to save intermediate files. When TRUE, the SnpSet object containing genotype calls is saved to the file indicated by 'crlmmFile'. This can be useful if an error occurs in the crlmmWrapper function after the genotyping has been completed.
`splitByChr`	Logical. Whether to split results by chromosome (default is TRUE). Helpful for datasets containing 50+ samples. Files are saved as `paste(dirname(crlmmFile), "crlmmSetList_", CHR, ".rda", sep="")`.

crlmmFile	Character string. Name of file (including the complete path) containing the genotype calls.
intensityFile	Character string. Name of file (with complete path) containing the quantile-normalized intensities.
rgFile	Character string. Name of file (including the complete path) containing the RG intensities (only applicable for illumina platform).
...	additional arguments to function readIdatFiles (illumina platform only)

Value

An object of class CrlmmSetList

Author(s)

R. Scharpf

See Also

CrlmmSetList crlmm

Examples

```
## Not run:
library(crlmm)
library(genomewidesnp6Crlmm)
celFiles <- list.files("PATH_TO_CELS", full.name=TRUE)
crlmmSetList <- crlmmWrapper(celFiles, intensityFile="intensities.rda")

## End(Not run)
```

---

list.celfiles *List CEL files.*

Description

Function used to get a list of CEL files.

Usage

```
list.celfiles(...)
```

Arguments

...	Same arguments of list.files

Details

For the moment, this function returns only uncompressed CEL files (ie, no CEL.gz)

Value

Character vector with filenames.

Note

Quite often users want to use this function to pass filenames to other methods. In this situations, it is safer to use the argument 'full.names=TRUE'.

See Also

list.files

Examples

```
if (require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## only the filenames
  list.celfiles(path)

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
  list.celfiles(path, full.names=TRUE)
}else{
  ## this won't return anything
  ## if in the working directory there isn't any CEL
  list.celfiles(getwd())
}
```

---

methods-eSet *Methods for eSet derivatives*

Description

`chromosome` return the chromosome number for SNP and NP probes.

`cnIndex, ...` return the row indices of copy number probes.

`position` return the physical position of the SNP or the physical position for the first index of the NP probe.

`snpIndex, ...` return the row indices of polymorphic loci.

Usage

```
chromosome(object)
cnIndex(object, ...)
position(object)
snpIndex(object, ...)
```

Arguments

`object`	Any object extending eSet (ABset, CopyNumberSet, CrlmmSetList, SnpSet
`...`	Usually the character string indicating annotation package.

Value

Vector of indices.

See Also

"ABset" "CopyNumberSet" "CrlmmSetList"

`calls` *Accessors for Calls and Confidences on a SnpSet object*

Description

`calls` returns the genotype calls. CRLMM stores genotype calls as integers (1 - AA; 2 - AB; 3 - BB).

`confs` returns the confidences associated to the genotype calls. THe current implementation of CRLMM stores the confidences as integers by using the transformation:

conf = round(-1000*log2(1-p)),

where 'p' is the posterior probability of the call.

Usage

```
calls(object)
confs(object)
```

Arguments

`object`	SnpSet object

Value

Matrix of genotype calls or confidences.

Examples

```
set.seed(1)
theCalls <- matrix(sample(1:3, 20, rep=TRUE), nc=2)
p <- matrix(runif(20), nc=2)
theConfs <- round(-1000*log2(1-p))
obj <- new("SnpSet", call=theCalls, callProbability=theConfs)
calls(obj)
confs(obj)
```

`readIdatFiles` *Reads Idat Files from Infinium II Illumina BeadChips*

Description

Reads intensity information for each bead type from .idat files of Infinium II genotyping BeadChips

Usage

```
readIdatFiles(sampleSheet=NULL, arrayNames=NULL, ids=NULL, path="",
              arrayInfoColNames=list(barcode="SentrixBarcode_A",
                                     position="SentrixPosition_A"),
              highDensity=FALSE, sep="_",
              fileExt=list(green="Grn.idat", red="Red.idat"),
              saveDate=FALSE)
```

Arguments

`sampleSheet`	`data.frame` containing Illumina sample sheet information (for required columns, refer to BeadStudio Genotyping guide - Appendix A).
`arrayNames`	character vector containing names of arrays to be read in. If `NULL`, all arrays that can be found in the specified working directory will be read in.
`ids`	vector containing ids of probes to be read in. If `NULL` all probes found on the first array are read in.
`path`	character string specifying the location of files to be read by the function
`arrayInfoColNames`	(used when `sampleSheet` is specified) list containing elements 'barcode' which indicates column names in the `sampleSheet` which contains the arrayNumber/barcode number and 'position' which indicates the strip number. In older style sample sheets, this information is combined (usually in a column named 'SentrixPosition') and this should be specified as `list(barcode=NULL, position="SentrixPosition")`
`highDensity`	logical (used when `sampleSheet` is specified). If `TRUE`, array extensions '_A', '_B' in sampleSheet are replaced with 'R01C01', 'R01C02' etc.
`sep`	character string specifying separator used in .idat file names.
`fileExt`	list containing elements 'Green' and 'Red' which specify the .idat file extension for the Cy3 and Cy5 channels.
`saveDate`	logical. Should the dates from each .idat be saved with sample information?

Details

The summarised Cy3 (G) and Cy5 (R) intensity, number of beads that were used in each channel and standard errors (all on the orginal scale) are read in from the .idat files.

Where available, a `sampleSheet` data.frame, in the same format as used by BeadStudio (columns 'Sample_ID', 'SentrixBarcode_A' and 'SentrixPosition_A' are required) which keeps track of sample information can be specified.

Thanks to Keith Baggerly who provided the code to read in the binary .idat files.

Value

NChannelSet with intensity data (R, G), number of beads (Rnb, Gnb) and standard errors (Rse, Gse) for each bead type.

Author(s)

Matt Ritchie

Examples

```
#RG = readIdatFiles()
```

---

snprma *Preprocessing tool for SNP arrays.*

Description

SNPRMA will preprocess SNP chips. The preprocessing consists of quantile normalization to a known target distribution and summarization to the SNP-Allele level.

Usage

```
snprma(filenames, mixtureSampleSize = 10^5, fitMixture = FALSE, eps = 0.1, verbo
```

Arguments

filenames	'character' vector with file names.
mixtureSampleSize	Sample size to be use when fitting the mixture model.
fitMixture	'logical'. Fit the mixture model?
eps	Stop criteria.
verbose	'logical'.
seed	Seed to be used when sampling.
cdfName	cdfName: 'GenomeWideSnp_5', 'GenomeWideSnp_6'
sns	Sample names.

Value

A	Summarized intensities for Allele A
B	Summarized intensities for Allele B
sns	Sample names
gns	SNP names
SNR	Signal-to-noise ratio
SKW	Skewness
mixtureParams	Parameters from mixture model
cdfName	Name of the CDF

Examples

```
if (require(genomewidesnp5Crlmm) & require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
  cels <- list.celfiles(path, full.names=TRUE)
  snprmaOutput <- snprma(cels)
  snprmaOutput[["A"]][1:10,]
  snprmaOutput[["B"]][1:10,]
}
```

Index

∗Topic **IO**
list.celfiles, 13
readIdatFiles, 16
∗Topic **classes**
ABset-class, 1
CopyNumberSet-class, 5
CrlmmSetList-class, 10
∗Topic **classif**
crlmm, 8
crlmmIllumina, 6
snprma, 17
∗Topic **list**
crlmmWrapper, 12
∗Topic **manip**
batch, 2
calls, 15
celDates, 2
computeCopynumber, 4
methods-eSet, 14
snprma, 17
∗Topic **methods**
crlmmWrapper, 12
∗Topic **models**
computeCopynumber, 4
∗Topic **package**
crlmm-package, 8
∗Topic **regression**
computeCopynumber, 4
∗Topic **robust**
cnrma, 3
computeCopynumber, 4
crlmmWrapper, 12
∗Topic **utilities**
list.celfiles, 13
[,CrlmmSetList-method
(CrlmmSetList-class), 10
$,CrlmmSetList-method
(CrlmmSetList-class), 10

A *(ABset-class)*, 1
A,ABset-method *(ABset-class)*, 1
A,CrlmmSetList-method
(CrlmmSetList-class), 10
A<- *(CrlmmSetList-class)*, 10
A<-,ABset,matrix-method
(ABset-class), 1
A<-,CrlmmSetList,matrix-method
(CrlmmSetList-class), 10
ABset, *12*, *15*
ABset-class, 1
addFeatureAnnotation
(CrlmmSetList-class), 10
addFeatureAnnotation,CrlmmSetList-method
(CrlmmSetList-class), 10
assayData, *10*

B *(ABset-class)*, 1
B,ABset-method *(ABset-class)*, 1
B,CrlmmSetList-method
(CrlmmSetList-class), 10
B<- *(CrlmmSetList-class)*, 10
B<-,ABset,matrix-method
(ABset-class), 1
B<-,CrlmmSetList,matrix-method
(CrlmmSetList-class), 10
batch, 2
batch,CopyNumberSet-method
(CopyNumberSet-class), 5
batch,CrlmmSetList-method
(CrlmmSetList-class), 10

CA *(CopyNumberSet-class)*, 5
CA,CopyNumberSet-method
(CopyNumberSet-class), 5
CA,CrlmmSetList-method
(CrlmmSetList-class), 10
CA<- *(CopyNumberSet-class)*, 5
CA<-,CopyNumberSet,matrix-method
(CopyNumberSet-class), 5
CA<-,CrlmmSetList,matrix-method
(CrlmmSetList-class), 10
calls, 15
calls,CrlmmSetList-method
(CrlmmSetList-class), 10
calls,SnpSet-method *(calls)*, 15
CB *(CopyNumberSet-class)*, 5
CB,CopyNumberSet-method
(CopyNumberSet-class), 5

CB,CrlmmSetList-method
(CrlmmSetList-class), 10
CB<- (CopyNumberSet-class), 5
CB<-,CopyNumberSet,matrix-method
(CopyNumberSet-class), 5
CB<-,CrlmmSetList,matrix-method
(CrlmmSetList-class), 10
celDates, 2
chromosome (methods-eSet), 14
chromosome,CopyNumberSet-method
(CopyNumberSet-class), 5
chromosome,CrlmmSetList-method
(CrlmmSetList-class), 10
chromosome,eSet-method
(methods-eSet), 14
cnIndex (methods-eSet), 14
cnIndex,CrlmmSetList-method
(CrlmmSetList-class), 10
cnIndex,eSet-method
(methods-eSet), 14
cnrma, 3
combine,CrlmmSetList,CrlmmSetList-method
(CrlmmSetList-class), 10
computeCopynumber, 4
confs (calls), 15
confs,CrlmmSetList-method
(CrlmmSetList-class), 10
confs,SnpSet-method (calls), 15
copyNumber,CopyNumberSet-method
(CopyNumberSet-class), 5
copyNumber,CrlmmSetList-method
(CrlmmSetList-class), 10
CopyNumberSet, 4, 15
CopyNumberSet-class, 5
crlmm, 8, 13
crlmm-package, 8
crlmmIllumina, 6
CrlmmSetList, 4, 13, 15
CrlmmSetList-class, 10
crlmmWrapper, 12, 12

ellipse,CopyNumberSet-method
(CopyNumberSet-class), 5
eSet, 1, 5

list, 10
list.celfiles, 13
list.files, 13, 14

methods-eSet, 14

ncol,CrlmmSetList-method
(CrlmmSetList-class), 10
nrow,CrlmmSetList-method
(CrlmmSetList-class), 10

plot,ABset,CopyNumberSet-method
(CopyNumberSet-class), 5
plot,CrlmmSetList,ANY-method
(CrlmmSetList-class), 10
points,CrlmmSetList-method
(CrlmmSetList-class), 10
position (methods-eSet), 14
position,CopyNumberSet-method
(CopyNumberSet-class), 5
position,CrlmmSetList-method
(CrlmmSetList-class), 10
position,eSet-method
(methods-eSet), 14
POSIXt, 3

read.celfile.header, 3
readIdatFiles, 16

sampleNames,CrlmmSetList-method
(CrlmmSetList-class), 10
show,CrlmmSetList-method
(CrlmmSetList-class), 10
snpIndex (methods-eSet), 14
snpIndex,CrlmmSetList-method
(CrlmmSetList-class), 10
snpIndex,eSet-method
(methods-eSet), 14
snprma, 17

update, 4
update,character-method
(CrlmmSetList-class), 10
update,CrlmmSetList-method
(CrlmmSetList-class), 10

vector, 10
Versioned, 1, 5
VersionedBiobase, 1, 5