oligo

November 11, 2009

R topics documented:

`basecontent` *Sequence Base Contents*

Description

Function to compute the amounts of each nucleotide in a sequence.

Usage

```
basecontent(seq)
```

Arguments

`seq`	character vector of length `n` containg a valid sequence (A/T/C/G)

Value

`matrix` with `n` rows and 4 columns with the counts for each base.

Examples

```
sequences <- c("ATATATCCCCG", "TTTCCGAGC")
basecontent(sequences)
```

basicRMA *Simplified Interface to RMA*

Description

Simple interface to RMA.

Usage

```
basicRMA(pmMat, pnVec, normalize = TRUE, background = TRUE, bgversion = 2, destr
```

Arguments

`pmMat`	Matrix of intensities to be processed.
`pnVec`	Probeset names.
`normalize`	Logical flag: normalize?
`background`	Logical flag: background adjustment?
`bgversion`	Version of background correction.
`destructive`	Logical flag: use destructive methods?
`verbose`	Logical flag: verbose.
`...`	Not currently used.

Value

Matrix.

Examples

```
set.seed(1)
pms <- matrix(rnorm(1000), nc=20)
colnames(pms) <- paste("sample", 1:20, sep="")
pns <- rep(letters[1:10], each=5)
res <- basicRMA(pms, pns, length(unique(pns)), TRUE, TRUE)
res[, 1:3]
```

`boxplot` *Boxplot*

Description

Boxplot for observed (log-)intensities in a FeatureSet-like object (ExpressionFeatureSet, ExonFeatureSet, SnpFeatureSet, TilingFeatureSet).

Usage

```
boxplot(x, ...)
## S4 method for signature 'FeatureSet':
boxplot(x, which=pmindex(x), transfo=log2,
range=0, ...)
## S4 method for signature 'ExpressionSet':
boxplot(x, which=1:nrow(x), transfo=identity,
range=0, ...)
```

Arguments

`x`	a `FeatureSet`-like object or `ExpressionSet` object.
`which`	an integer vector determining which rows of 'x' should be plotted. See 'Details'.
`transfo`	a function to transform the data before plotting. See 'Details'.
`range`	this determines how far the plot whiskers extend out from the box. If 'range' is positive, the whiskers extend to the most extreme data point which is no more than 'range' times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
`...`	arguments to be passed to plot

Details

The 'which' argument should be used to subset the object to be plotted. For example, if the user wants to plot the PM probes, he should use 'which=pmindex(x)', if MM probes are to be plotted 'which=mmindex(x)'. If all probes are to be plotted 'which=1:nrow(x)'. Note that pmindex/mmindex options will not work for summarized data.

The 'transfo' argument will set the transformation to be used. For raw data, 'transfo=log2' is a common practice. For summarized data (which are often in log2-scale), no transformation is needed (therefore 'transfo=identity').

See Also

`hist`, `image`

getX *Accessors for physical array coordinates.*

Description

Accessors for physical array coordinates.

Usage

```
getX(object, type)
getY(object, type)
```

Arguments

`object`	`FeatureSet` object
`type`	'character' defining the type of the probes to be queried. Valid options are 'pm', 'mm', 'bg'

Value

A vector with the requested coordinates.

Examples

```
## Not run:
x <- read.celfiles(list.celfiles())
theXpm <- getX(x, "pm")
theYpm <- getY(x, "pm")
## End(Not run)
```

crlmm *Genotype Calls*

Description

Performs genotype calls via CRLMM (Corrected Robust Linear Model with Maximum-likelihood based distances).

Usage

```
crlmm(filenames, outdir, batch_size=40000, balance=1.5,
      minLLRforCalls=c(5, 1, 5), recalibrate=TRUE,
      verbose=TRUE, pkgname, reference=TRUE)
justCRLMM(filenames, batch_size = 40000, minLLRforCalls = c(5, 1, 5),
recalibrate = TRUE, balance = 1.5, phenoData = NULL, verbose = TRUE,
pkgname = NULL, tmpdir=tempdir())
```

Arguments

`filenames` character vector with the filenames.

`outdir` directory where the output (and some tmp files) files will be saved.

`batch_size` integer defining how many SNPs should be processed at a time.

`recalibrate` Logical - should recalibration be performed?

`balance` Control parameter to balance homozygotes and heterozygotes calls.

`minLLRforCalls` Minimum thresholds for genotype calls.

`verbose` Logical.

`phenoData` `phenoData` object or `NULL`

`pkgname` alt. pdInfo package to be used

`reference` logical, defaulting to TRUE ...

`tmpdir` Directory where temporary files are going to be stored at.

Value

`SnpCallSetPlus` object.

`getContainer` *Get container information for NimbleGen Tiling Arrays.*

Description

Get container information for NimbleGen Tiling Arrays. This is useful for better identification of control probes.

Usage

```
getContainer(object, probeType)
```

Arguments

`object` A `TilingFeatureSet` or `TilingFeatureSet2` object.

`probeType` String describing which probes to query ('pm', 'bg')

Value

'character' vector with container information.

`getCrlmmSummaries` *Function to get CRLMM summaries saved to disk*

Description

This will read the summaries written to disk and return them to the user as a `SnpCallSetPlus` or `SnpCnvCallSetPlus` object.

Usage

```
getCrlmmSummaries(tmpdir)
```

Arguments

`tmpdir`	directory where CRLMM saved the results to.

Value

If the data were from SNP 5.0 or 6.0 arrays, the function will return a `SnpCnvCallSetPlus` object. It will return a `SnpCallSetPlus` object, otherwise.

`getNgsColorsInfo` *Helper function to extract color information for filenames on NimbleGen arrays.*

Description

This function will (try to) extract the color information for NimbleGen arrays. This is useful when using `read.xysfiles2` to parse XYS files for Tiling applications.

Usage

```
getNgsColorsInfo(path = ".", pattern1 = "_532", pattern2 = "_635", ...)
```

Arguments

`path`	path where to look for files
`pattern1`	pattern to match files supposed to go to the first channel
`pattern2`	pattern to match files supposed to go to the second channel
`...`	extra arguments for `list.xysfiles`

Details

Many NimbleGen samples are identified following the pattern sampleID_532.XYS / sampleID_635.XYS.

The function suggests sample names if all the filenames follow the standard above.

Value

A data.frame with, at least, two columns: 'channel1' and 'channel2'. A third column, 'sampleNames', is returned if the filenames follow the sampleID_532.XYS / sampleID_635.XYS standard.

Author(s)

Benilton Carvalho <bcarvalh@jhsph.edu>

`hist` *Density estimate*

Description

Plot the density estimates for each sample

Usage

```
hist(x, ...)
```

Arguments

`x`	`FeatureSet` object
`...`	arguments to be passed to `lines`

`image` *Display a Color Image*

Description

Produces an image (`graphics::image`) for each sample.

Usage

```
image(x, ...)
```

Arguments

`x`	`FeatureSet` object
`...`	parameters to be passed to `plot`

`mm` *Accessors and replacement methods for the PM/MM/BG matrices.*

Description

Accessors and replacement methods for the PM/MM/BG matrices.

Usage

```
mm(object, subset = NULL)
pm(object, subset = NULL, ...)
bg(object, subset = NULL)
mm(object)<-value
pm(object)<-value
bg(object)<-value
```

Arguments

`object`	`FeatureSet` object.
`subset`	Not implemented yet.
`value`	`matrix` object.
`...`	Extra arguments for future implementation.

Details

For all objects but `TilingFeatureSet2`, these methods will return matrices. In case of `TilingFeatureSet2` objects, the value is a 3-dimensional array (probes x samples x channels).

Examples

```
if (require(maqcExpression4plex) & require(pd.hg18.60mer.expr)){
xysPath <- system.file("extdata", package="maqcExpression4plex")
xysFiles <- list.xysfiles(xysPath, full.name=TRUE)
ngsExpressionFeatureSet <- read.xysfiles(xysFiles)
pm(ngsExpressionFeatureSet)[1:10,]
}
```

`justSNPRMA` *Summarization of SNP data*

Description

This function implements the SNPRMA method for summarization of SNP data. It works directly with the CEL files, saving memory.

Usage

```
justSNPRMA(filenames, verbose = TRUE, phenoData = NULL, normalizeToHapmap = TRUE
```

Arguments

`filenames`	character vector with the filenames.
`verbose`	logical flag for verbosity.
`phenoData`	a `phenoData` object or `NULL`
`normalizeToHapmap`	Normalize to Hapmap? Should always be TRUE, but it's kept here for future use.

Value

`SnpQSet` or a `SnpCnvQSet`, depending on the array type.

Examples

```
## snprmaResults <- justSNPRMA(list.celfiles())
```

`list.celfiles` *List CEL/XYS files*

Description

Lists the CEL/XYS files.

Usage

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

Arguments

`...`	parameters to be passed to `list.files`

Value

Character vector with the filenames.

Examples

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

`MAplot-methods` *MA plots*

Description

Create MA plots using a reference array (based on medians).

Methods

object = "FeatureSet" ExpressionFeatureSet

oligo-package *The oligo package: a tool for low-level analysis of oligonucleotide arrays*

Description

The **oligo** package handles oligonucleotide arrays: expression, tiling, SNP and exon chips. The supported manufacturers are Affymetrix and NimbleGen. The package provides tools for preprocessing.

Details

The package will read the raw intensity files (CEL for Affymetrix; XYS for NimbleGen) and allow the user to perform analyses starting at the feature-level.

Reading in the intensity files require the existence of data packages that contain the chip specific information (X/Y coordinates; feature types; sequence). These data packages packages are built using the **pdInfoBuilder** package.

For Affymetrix SNP arrays, users are asked to download the already built annotation packages from BioConductor. This is because these packages contain metadata that are not (yet) automatically created. The following annotation packages are available:

50K Xba - pd.mapping50kxba.240 50K Hind - pd.mapping50khind.240 250K Sty - pd.mapping250k.sty 250K Nsp - pd.mapping250k.nsp GenomeWideSnp 5 (SNP 5.0) - pd.genomewidesnp.5 GenomeWideSnp 6 (SNP 6.0) - pd.genomewidesnp.6

For users interested in genotype calls for SNP 5.0 and 6.0 arrays, we strongly recommend the use use the **crlmm** package, which implements a more efficient version of CRLMM.

Author(s)

Benilton Carvalho - ⟨bcarvalh@jhsph.edu⟩

References

Carvalho, B.; Bengtsson, H.; Speed, T. P. & Irizarry, R. A. Exploration, Normalization, and Genotype Calls of High Density Oligonucleotide SNP Array Data. Biostatistics, 2006.

plotM-methods *Methods for Log-Ratio plotting*

Description

The `plotM` methods are meant to plot log-ratios for different classes of data.

Methods

object = "SnpQSet", i = "character" Plot log-ratio for SNP data for sample i.

object = "SnpQSet", i = "integer" Plot log-ratio for SNP data for sample i.

object = "SnpQSet", i = "numeric" Plot log-ratio for SNP data for sample i.

object = "TilingQSet", i = "missing" Plot log-ratio for Tiling data for sample i.

read.celfiles *Parser to CEL files*

Description

Reads CEL files.

Usage

```
read.celfiles(..., filenames, pkgname, phenoData, featureData,
experimentData, notes, verbose = TRUE, sampleNames, rm.mask = FALSE,
rm.outliers = FALSE, rm.extra = FALSE, sd = FALSE, checkType = TRUE,
useAffyio = TRUE)

read.celfiles2(channel1, channel2, pkgname, phenoData, featureData,
experimentData, notes, verbose = TRUE, sampleNames, rm.mask = FALSE,
rm.outliers = FALSE, rm.extra = FALSE, sd = FALSE, checkType = TRUE,
useAffyio = TRUE)
```

Arguments

`...` names of files to be read.

`filenames` a `character` vector with the CEL filenames.

`channel1` a `character` vector with the CEL filenames for the first 'channel' on a Tiling application

`channel2` a `character` vector with the CEL filenames for the second 'channel' on a Tiling application

`pkgname` alternative data package to be loaded.

`phenoData` `phenoData`

`featureData` `featureData`

`experimentData` `experimentData`

`notes` `notes`

`verbose` `logical`

`sampleNames` `character` vector with sample names (usually better descriptors than the filenames)

`rm.mask` `logical`. Read masked?

`rm.outliers` `logical`. Remove outliers?

`rm.extra` `logical`. Remove extra?

`sd` `logical`. Read SD?

`checkType` `logical`. Check type of each file? This can be time consuming.

`useAffyio` `logical`. Use 'affyio' instead of 'affxparser' to read in CEL files.

Details

When using 'affyio' to read in CEL files, the user can read compressed CEL files (CEL.gz). Additionally, 'affyio' is much faster than 'affxparser'.

Value

`this-is-escaped-codenormal-bracket58bracket-normal`
: if Expresssion arrays

`this-is-escaped-codenormal-bracket61bracket-normal`
: if Exon arrays

`this-is-escaped-codenormal-bracket64bracket-normal`
: if SNP arrays

`this-is-escaped-codenormal-bracket67bracket-normal`
: if Tiling arrays

See Also

`list.celfiles`, `read.xysfiles`

Examples

```
if(require(pd.mapping50k.xba240) & require(hapmap100kxba)){
celPath <- system.file("celFiles", package="hapmap100kxba")
celFiles <- list.celfiles(celPath, full.name=TRUE)
affySnpFeatureSet <- read.celfiles(celFiles)
}
```

`readSummaries` *Read summaries generated by crlmm*

Description

This function read the different summaries generated by crlmm.

Usage

```
readSummaries(type, tmpdir)
```

Arguments

`type`	type of summary of `character` class: 'alleleA', 'alleleB', 'alleleA-sense', 'alleleA-antisense', 'alleleB-sense', 'alleleB-antisense', 'calls', 'llr', 'conf'.
`tmpdir`	directory containing the output saved by crlmm

Details

On the 50K and 250K arrays, given a SNP, there are probes on both strands (sense and antisense). For this reason, the options 'alleleA-sense', 'alleleA-antisense', 'alleleB-sense' and 'alleleB-antisense' should be used **only** with such arrays (XBA, HIND, NSP or STY).

On the SNP 5.0 and SNP 6.0 platforms, this distinction does not exist in terms of algorithm (note that the actual strand could be queried from the annotation package). For these arrays, options 'alleleA', 'alleleB' are the ones to be used.

The options `calls`, `llr` and `conf` will return, respectively, the CRLMM calls, log-likelihood ratios (for devel purpose **only**) and CRLMM confidence calls matrices.

Value

Matrix with values of summaries.

read.xysfiles *Parser to XYS files*

Description

NimbleGen provides XYS files which are read by this function.

Usage

```
read.xysfiles(..., filenames, pkgname, phenoData, featureData,
experimentData, notes, verbose=TRUE, sampleNames, checkType=TRUE)

read.xysfiles2(channel1, channel2, pkgname, phenoData, featureData,
experimentData, notes, verbose=TRUE, sampleNames, checkType=TRUE)
```

Arguments

`...`	file names
`filenames`	`character` vector with filenames.
`channel1`	a `character` vector with the CEL filenames for the first 'channel' on a Tiling application
`channel2`	a `character` vector with the CEL filenames for the second 'channel' on a Tiling application
`pkgname`	`character` vector with alternative PD Info package name
`phenoData`	`phenoData`
`featureData`	`featureData`
`experimentData`	`experimentData`
`notes`	`notes`
`verbose`	`verbose`
`sampleNames`	`character` vector with sample names (usually better descriptors than the filenames)
`checkType`	`logical`. Check type of each file? This can be time consuming.

Details

The function will read the XYS files provided by NimbleGen Systems and return an object of class FeatureSet.

Value

`this-is-escaped-codenormal-bracket44bracket-normal`
if Expresssion arrays

`this-is-escaped-codenormal-bracket47bracket-normal`
if Tiling arrays

See Also

`list.xysfiles`, `read.celfiles`

Examples

```
if (require(maqcExpression4plex) & require(pd.hg18.60mer.expr)){
xysPath <- system.file("extdata", package="maqcExpression4plex")
xysFiles <- list.xysfiles(xysPath, full.name=TRUE)
ngsExpressionFeatureSet <- read.xysfiles(xysFiles)
}
```

`rma` *RMA - Robust Multichip Average algorithm*

Description

Robust Multichip Average methodology. This will convert an (Expression/Exon/Gene)FeatureSet object to an ExpressionSet object by using RMA strategy.

Usage

```
rma(object, ...)
```

Arguments

`object`	`FeatureSet` object
`...`	Extra arguments.

Details

This function computes the RMA (Robust Multichip Average) expression measure described in Irizarry et al Biostatistics (2003).

Note that this expression measure is given to you in log base 2 scale. This differs from most of the other expression measure methods.

For Exon ST and Gene ST arrays, the user should be aware that the summarization is performed to the *probeset* level. The ExpressionSet returned when either Exon/Gene-FeatureSet objects are passed contain extra annotation on the featureData slot that the user should take into account for exon/gene-level analyses.

Value

`ExpressionSet` object.

References

Rafael. A. Irizarry, Benjamin M. Bolstad, Francois Collin, Leslie M. Cope, Bridget Hobbs and Terence P. Speed (2003), Summa ries of Affymetrix GeneChip probe level data Nucleic Acids Research 31(4):e15

Bolstad, B.M., Irizarry R. A., Astrand M., and Speed, T.P. (2003), A Comparison of Normalization Methods for High Density O ligonucleotide Array Data Based on Bias and Variance. Bioinformatics 19(2):185-193

Irizarry, RA, Hobbs, B, Collin, F, Beazer-Barclay, YD, Antonellis, KJ, Scherf, U, Speed, TP (2003) Exploration, Normalizati on, and Summaries of High Density Oligonucleotide Array Probe Level Data. Biostatistics .Vol. 4, Number 2: 249-264

See Also

snprma

Examples

```
if (require(maqcExpression4plex) & require(pd.hg18.60mer.expr)){
xysPath <- system.file("extdata", package="maqcExpression4plex")
xysFiles <- list.xysfiles(xysPath, full.name=TRUE)
ngsExpressionFeatureSet <- read.xysfiles(xysFiles)
summarized <- rma(ngsExpressionFeatureSet)
show(summarized)
}
```

---

sequenceDesignMatrix

Create design matrix for sequences

Description

Creates design matrix for sequences.

Usage

```
sequenceDesignMatrix(seqs)
```

Arguments

`seqs`	`character` vector of 25-mers.

Details

This assumes all sequences are 25bp long.

The design matrix is often used when the objective is to adjust intensities by sequence.

Value

Matrix with length(seqs) rows and 75 columns.

Examples

```
genSequence <- function(x)
    paste(sample(c("A", "T", "C", "G"), 25, rep=TRUE), collapse="", sep="")
seqs <- sapply(1:10, genSequence)
X <- sequenceDesignMatrix(seqs)
Y <- rnorm(10, mean=12, sd=2)
Ydemean <- Y-mean(Y)
X[1:10, 1:3]
fit <- lm(Ydemean~X)
coef(fit)
```

`snprma` *Preprocessing SNP Arrays*

Description

This function preprocess SNP arrays.

Usage

```
snprma(object, verbose = TRUE, normalizeToHapmap = TRUE)
```

Arguments

`object`	`SnpFeatureSet` object
`verbose`	Verbosity flag. `logical`
`normalizeToHapmap`	internal

Value

A `SnpQSet` object.

Index