Package ‘NanoStringQCPro’

January 31, 2017

Title Quality metrics and data processing methods for NanoString mRNA gene expression data

Version 1.6.0

Date 2016-03-03

Author Dorothee Nickles <nickles.dorothee@gene.com>, Thomas Sandmann <sandmann.thomas@gene.com>, Robert Ziman <ziman.robert@gene.com>, Richard Bourgon <bourgon.richard@gene.com>

Maintainer Robert Ziman <ziman.robert@gene.com>

Description NanoStringQCPro provides a set of quality metrics that can be used to assess the quality of NanoString mRNA gene expression data -- i.e. to identify outlier probes and outlier samples. It also provides different background subtraction and normalization approaches for this data. It outputs suggestions for flagging samples/probes and an easily sharable html quality control output.

Depends R (&gt;= 3.2), methods

Imports AnnotationDbi (&gt;= 1.26.0), org.Hs.eg.db (&gt;= 2.14.0), Biobase (&gt;= 2.24.0), knitr (&gt;= 1.12), NMF (&gt;= 0.20.5), RColorBrewer (&gt;= 1.0-5), png (&gt;= 0.1-7)

Suggests roxygen2 (&gt;= 4.0.1), testthat, BiocStyle

License Artistic-2.0

biocViews Microarray, mRNAMicroarray, Preprocessing, Normalization, QualityControl, ReportWriting

LazyData true

NeedsCompilation no

R topics documented:

addCodesetAnnotation,RccSet-method ........................................... 3
addQCFflags,RccSet-method .................................................. 4
allSumPlot,RccSet-method .................................................. 4
assessHousekeeping,RccSet-method .......................................... 5
bdPlot,RccSet-method ........................................................ 6
buildCodesetAnnotation ....................................................... 6
R topics documented:

<table>
<thead>
<tr>
<th>Function Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkRccSet,RccSet-method</td>
<td>7</td>
</tr>
<tr>
<td>colByCovar</td>
<td>8</td>
</tr>
<tr>
<td>colByFun</td>
<td>9</td>
</tr>
<tr>
<td>contentNorm,RccSet-method</td>
<td>9</td>
</tr>
<tr>
<td>copyRccSet,RccSet-method</td>
<td>10</td>
</tr>
<tr>
<td>countsInBlankSamples_verticalPlot</td>
<td>11</td>
</tr>
<tr>
<td>ctrlsOverviewPlot,RccSet-method</td>
<td>12</td>
</tr>
<tr>
<td>ctrlsZprimePlot,RccSet-method</td>
<td>12</td>
</tr>
<tr>
<td>cutoffByMMAD</td>
<td>13</td>
</tr>
<tr>
<td>cutoffByVar</td>
<td>13</td>
</tr>
<tr>
<td>dCoVar</td>
<td>14</td>
</tr>
<tr>
<td>densityPlot</td>
<td>14</td>
</tr>
<tr>
<td>example_rccSet</td>
<td>15</td>
</tr>
<tr>
<td>flagSamplesCount,RccSet-method</td>
<td>15</td>
</tr>
<tr>
<td>flagSamplesCtrl,RccSet-method</td>
<td>16</td>
</tr>
<tr>
<td>flagSamplesTech,RccSet-method</td>
<td>17</td>
</tr>
<tr>
<td>fovPlot,RccSet-method</td>
<td>17</td>
</tr>
<tr>
<td>geneClustering</td>
<td>18</td>
</tr>
<tr>
<td>getBackground,RccSet-method</td>
<td>18</td>
</tr>
<tr>
<td>getBlankLabel,RccSet-method</td>
<td>20</td>
</tr>
<tr>
<td>getSpikeInInput</td>
<td>20</td>
</tr>
<tr>
<td>iqrPlot,RccSet-method</td>
<td>21</td>
</tr>
<tr>
<td>lodAssess,RccSet-method</td>
<td>22</td>
</tr>
<tr>
<td>lodPlot,RccSet-method</td>
<td>22</td>
</tr>
<tr>
<td>makeQCReport,RccSet-method</td>
<td>23</td>
</tr>
<tr>
<td>myCols</td>
<td>24</td>
</tr>
<tr>
<td>NanoStringQCPro</td>
<td>25</td>
</tr>
<tr>
<td>negCtrlsByLane,RccSet-method</td>
<td>25</td>
</tr>
<tr>
<td>negCtrlsByLane_verticalPlot</td>
<td>26</td>
</tr>
<tr>
<td>negCtrlsPairs,RccSet-method</td>
<td>26</td>
</tr>
<tr>
<td>negCtrlsPlot,RccSet-method</td>
<td>27</td>
</tr>
<tr>
<td>newRccSet</td>
<td>27</td>
</tr>
<tr>
<td>nSolverBackground,RccSet-method</td>
<td>29</td>
</tr>
<tr>
<td>nSolverCsv.to.pdata_fdata_adata</td>
<td>31</td>
</tr>
<tr>
<td>panelCor</td>
<td>31</td>
</tr>
<tr>
<td>pcaPlot</td>
<td>32</td>
</tr>
<tr>
<td>pdata_fdata_adata.to.rccSet</td>
<td>32</td>
</tr>
<tr>
<td>posCtrlNorm,RccSet-method</td>
<td>33</td>
</tr>
<tr>
<td>posNormFacPlot,RccSet-method</td>
<td>34</td>
</tr>
<tr>
<td>posR2Plot,RccSet-method</td>
<td>34</td>
</tr>
<tr>
<td>posRatioPlot,RccSet-method</td>
<td>35</td>
</tr>
<tr>
<td>posSlopePlot,RccSet-method</td>
<td>36</td>
</tr>
<tr>
<td>posSumVsAllSumPlot,RccSet-method</td>
<td>36</td>
</tr>
<tr>
<td>preprocRccSet,RccSet-method</td>
<td>37</td>
</tr>
<tr>
<td>presAbsCall,RccSet-method</td>
<td>39</td>
</tr>
<tr>
<td>previewPNG</td>
<td>40</td>
</tr>
<tr>
<td>rccFiles.to.pdata_fdata_adata</td>
<td>41</td>
</tr>
<tr>
<td>RccSet</td>
<td>41</td>
</tr>
<tr>
<td>RccSet-class</td>
<td>42</td>
</tr>
<tr>
<td>readCdrDesignData</td>
<td>43</td>
</tr>
<tr>
<td>readRcc</td>
<td>44</td>
</tr>
<tr>
<td>readRccBatch</td>
<td>44</td>
</tr>
</tbody>
</table>
addCodesetAnnotation,RccSet-method

Add NanoString codeset annotation to an RccSet

Description

Returns a copy of the input RccSet where the codeset annotation has been merged into its fData slot. The merge key for each is a string formed from the concatenation of their CodeClass, GeneName, and Accession columns ("<CodeClass>_<GeneName>_<Accession>"). For creating the codeset annotation object, see buildCodesetAnnotation().

Usage

## S4 method for signature 'RccSet'
addCodesetAnnotation(rccSet, annot, reorder = TRUE,
showWarnings = TRUE)

Arguments

rccSet An RccSet object.
annot Data frame containing the codeset annotation.
reorder Logical indicating whether the probes should be reordered according to their barcodes (this can help in identifying barcode-specific artifacts – i.e. background noise).
showWarnings Logical indicating whether or not warnings should be shown, if any.

Value

A copy of the input RccSet where the codeset annotation has been merged into its fData slot.

Author(s)

Dorothee Nickles, Robert Ziman

Examples

rccDir <- system.file("extdata", "RCC", package="NanoStringQCPro")
rccSet <- newRccSet(rccFiles = dir(rccDir, full.names=TRUE))
rlf <- system.file("extdata", "RLF", "NQCP_example.rlf", package="NanoStringQCPro")
annot <- buildCodesetAnnotation(rlf)
rccSet.annotated <- addCodesetAnnotation(rccSet, annot)
Add sample QC flags to an RccSet

Description

Returns a copy of the input RccSet with columns added to pData from the provided sample QC flag annotation file. (That file is produced by makeQCReport(); see its help page for more details.)

Usage

## S4 method for signature 'RccSet'
addQCFlags(rccSet, flagFile)

Arguments

rccSet  An RccSet object
flagFile Path to a sample QC flag file as generated by the NanoStringQCPro QC report (see makeQCReport())

Value

A copy of the input RccSet with columns added to pData from the QC flag file.

Author(s)

Dorothee Nickles

Plot the sum of all counts (endogenous and housekeeping genes only) for each sample in an RccSet object.

Usage

## S4 method for signature 'RccSet'
allSumPlot(rccSet, method = c("cutoffByMMAD", "cutoffByVar"), stringency = 4)

Arguments

rccSet  An RccSet object
method Character string specifying the method for outlier detection: either "cutoffByMMAD" or "cutoffByVar".
stringency Numeric value passed to the cutoff function specified by the method argument (see the ‘d’ argument of cutoffByMMAD and cutoffByVar).
assessHousekeeping,RccSet-method

Details
The sum of counts for each sample in the RccSet are plotted and and outliers (as determined the cut-off function specified by the method argument) are marked in red (thresholds for outlier definition are plotted as red dashed lines).

Value
A plot

Author(s)
Dorothee Nickles

assessHousekeeping,RccSet-method

assessHousekeeping

Description
Assess correlation and variance/variability of housekeeping genes

Usage
## S4 method for signature 'RccSet'
assessHousekeeping(rccSet, hk, covar, annotate = TRUE,
plot = TRUE, digits = 2)

Arguments
rccSet An RccSet object
hk Either a boolean vector of length nrow(exprs(rccSet)) or a numeric vector of
indices which genes in exprs(rccSet) are housekeeping genes
covar character; colname in fData(rccSet) that can be used to label genes by a category
of interest
annotate Scalar boolean; if TRUE (default), probes will be "annotated" using the "Gene-
Name" column in the fData(rccSet) slot
plot Scalar boolean, plot pairwise relationships ?
digits Scalar integer, the number of decimal places

Details
Pairwise correlations of all defined housekeeping genes will be assessed and pairwise scatterplots
will be generated. This function does not only output pairwise correlation coefficients, but also - for
each housekeeping gene - the variance, the interquartile range (IQR) and median expression level
across all samples in the experiment.

Value
A dataframe with one row per housekeeping genes and several columns with metrics suggested to
assess performance of defined housekeeping genes.
bdPlot,RccSet-method  

**Binding density plot**

**Description**

Plot the binding density of each sample in an RccSet object. Samples with a binding density < 0.05 or > 2.25 (thresholds defined by NanoString) are marked in red (dashed red line indicates threshold).

**Usage**

```r
## S4 method for signature 'RccSet'
bdPlot(rccSet)
```

**Arguments**

- `rccSet`  
  An RccSet object

**Value**

A plot

**Author(s)**

Dorothee Nickles

---

**buildCodesetAnnotation**

*Build NanoString codeset annotation*

**Description**

This function returns a data frame whose content is the combination of the NanoString-provided codeset annotation (.RLF file and the "Design Data" tab of the CDR spreadsheet) with gene annotation in the org.Hs.eg.db package.

**Usage**

```r
buildCodesetAnnotation(rlf = NULL, cdrDesignData = NULL, removeRedundantCols = TRUE, addEgAnnotations = FALSE)
```
Arguments

rlf Path to the RLF file
cdrDesignData Path to a manually prepared .CSV export of the "Design Data" tab of the CDR file (optional; see 'details' section below for how the export should be prepared)
removeRedundantCols Logical. If TRUE, cols in the CDR that are redundant with those in the RLF will be omitted from the output.
addEgAnnotations Logical indicating whether or not to add EntrezGene IDs and HGNC symbols from the org.Hs.eg.db package.

Details

The original NanoString provided .RLF file is expected as input. This file is the master (i.e. only probes listed here will be annotated; any extra ones in the CDR export will be dropped). If the CDR "Design Data" .CSV is specified, the function expects this .CSV file to be generated from the "Design Data" tab of the original NanoString provided Excel CDR file. This tab needs to be trimmed by skipping the NanoString header and first column containing only integers; the resulting .CSV should contain the actual table (including its header – beginning with "Customer Identifier"). The function will match and join the .RLF and CDR .CSV using their "ProbeID" and "NSID" fields, and then it will add gene annotation (EntrezGene ID, HGNC symbol, and chromosomal position) by doing lookups in the org.Hs.eg.db package using the RefSeq accessions from the RLF.

Value

A data frame whose content is the combination of the NanoString-provided codeset annotation with gene annotation in the org.Hs.eg.db package.

Author(s)

Dorothee Nickles, Robert Ziman

Examples

rlf <- system.file("extdata", "RLF", "NQCP_example.rlf", package="NanoStringQCPro")
cdrDesignData <- system.file("extdata", "CDR", "CDR-DesignData.csv", package="NanoStringQCPro")
annot <- buildCodesetAnnotation(rlf, cdrDesignData)

checkRccSet, RccSet-method

Check an RccSet

Description

Provides additional checks and generates warnings for unexpected or unusual conditions which, though permitted by the RccSet class, may indicate data import errors.

Usage

# S4 method for signature 'RccSet'
checkRccSet(rccSet, reportWarnings = TRUE,
            showMessages = FALSE)
colByCovar

Arguments

- `rccSet` An RccSet to be checked.
- `reportWarnings` Logical. If TRUE, warnings are reported.
- `showMessages` Logical. If TRUE, notes are shown indicating any optional missing columns and the like.

Value

Returns TRUE if no warnings were generated and FALSE otherwise.

Author(s)

Robert Ziman

Examples

```r
data(example_rccSet)
checkRccSet(example_rccSet)
```

colByCovar

Description

Define colors based on a covariate of an RccSet object

Usage

```r
colByCovar(pdata, covar)
```

Arguments

- `pdata` pData() of an RccSet object
- `covar` character, colname in the pdata used to stratify (color) data

Value

A list of length 2, with ["color"] being a character vector of colors (one color for each level of covar) of length=number of observations and ["legend"] providing the levels of covar to map colors to covar

Author(s)

Dorothee Nickles
**Description**

Color `x` based on upper and lower thresholds

**Usage**

`colByFun(x, thresholds)`

**Arguments**

- `x`: Numeric vector
- `thresholds`: List of length 2, with a scalar numeric in each slot, one giving the lower the upper threshold (for outlier definition)

**Value**

A vector of colors, with "red" for all values of `x` exceeding thresholds and "black" for all other values

**Author(s)**

Dorothee Nickles

---

**contentNorm, RccSet-method**

*Content normalization*

**Description**

Performs content normalization on the given RccSet.

**Usage**

```r
## S4 method for signature 'RccSet'
contentNorm(rccSet, method = c("global", "housekeeping"),
            summaryFunction = "median", hk = NULL, inputMatrix = c("bgCorrData",
            "posCtrlData", "exprs"), quietly = FALSE)
```

**Arguments**

- `rccSet`: An RccSet.
- `method`: Specifies the features to be used for normalization. "global" indicates that all features should be used and "housekeeping" indicates that only housekeeping features should be used. If "housekeeping" is specified and the 'hk' argument (below) is also specified, then the features indicated by 'hk' will be used. If "housekeeping" is specified and 'hk' is left NULL, then the default housekeeping features (i.e. those with CodeClass == "Housekeeping") will be used.
copyRccSet.RccSet-method

Deep-copy a NanoString RccSet

Description

Returns a copy of the input RccSet where the copy's assayData has been produced via copyEnv() rather than a simple assignment – hence deep-copying the environment pointed to by assayData rather than just copying the pointer. This guarantees that if the copy's assayData is affected later in the code, assayData for the original won't be affected.
### copyRccSet

#### Usage

```r
## S4 method for signature 'RccSet'
copyRccSet(rccSet)
```

#### Arguments

- `rccSet`: A NanoString RccSet to be copied.

#### Value

A new RccSet that is a deep copy of the original.

#### Author(s)

Robert Ziman

#### Examples

```r
data(example_rccSet)
example_rccSet_2 <- copyRccSet(example_rccSet)
assayData(example_rccSet)
assayData(example_rccSet_2) # Should be different
```

---

### countsInBlankSamples_verticalPlot

**Plot counts in blank samples (vertical orientation)**

#### Usage

```r
countsInBlankSamples_verticalPlot(rccSet, outputFile)
```

#### Arguments

- `rccSet`: An RccSet
- `outputFile`: Output PNG filename

#### Value

A PNG file containing a boxplot of the gene-wise counts for the blank samples in the input. The PNG is set to a fixed resolution of 300 pixels per inch and a fixed width of 2250 pixels (i.e. 7.5” at 300ppi), but the height varies with the size of the input. The font size is also fixed so that the labels will be legible even for large datasets.
**Description**

Plot individual negative and positive controls across all samples in an RccSet object.

**Usage**

```r
## S4 method for signature 'RccSet'
ctrlsOverviewPlot(rccSet)
```

**Arguments**

- `rccSet` An RccSet object

**Value**

A plot with two panels, one for the negative controls, one for the positive controls.

**Author(s)**

Dorothee Nickles

---

**Description**

Plot distribution of counts and the Z' Factors comparing the negative controls and the three highest input positive controls of an RccSet object.

**Usage**

```r
## S4 method for signature 'RccSet'
ctrlsZprimePlot(rccSet)
```

**Arguments**

- `rccSet` An RccSet object

**Value**

A plot

**Author(s)**

Dorothee Nickles
**Description**

Determine cutoffs of x (for outlier detection) based on median

**Usage**

`cutoffByMMAD(x, d, ...)`

**Arguments**

- `x`: numeric vector
- `d`: scalar numeric, factor by which to multiply MAD of x
- `...`: additional parameters passed on to `median()`

**Value**

A list of length 2, with a scalar numeric in each slot, one giving the lower threshold (median(x) - d * mad(x)), the other giving the upper threshold (median(x) + d * mad(x)) for outlier definition.

**Author(s)**

Dorothee Nickles

---

**Description**

Determine cutoffs of x (for outlier detection) based on a certain percent CV

**Usage**

`cutoffByVar(x, d, ...)`

**Arguments**

- `x`: numeric vector
- `d`: scalar numeric, percent CV; passed on to `dCoVar`
- `...`: additional parameters passed on to `mean()`

**Value**

A list of length 2, with a scalar numeric in each slot, one giving the lower threshold (mean(x) - CV the other giving the upper threshold (mean(x) + percent CV based cutoff) for outlier definition.

**Author(s)**

Dorothee Nickles
**dCoVar**

**Description**

Determine standard deviation at a certain percent CV

**Usage**

dCoVar(x, d, ...)

**Arguments**

- **x** numeric vector
- **d** scalar numeric, percent CV
- **...** additional parameters passed on to mean()

**Value**

standard deviation of x at d percent of CV

**Author(s)**

Dorothee Nickles

---

**densityPlot**

**Description**

Plot the density of counts for all endogenous and housekeeping genes for each sample in an RccSet object

**Usage**

densityPlot(M, log.transform = FALSE, pdata, covar, ...)

**Arguments**

- **M** One of the matrices from the assayData() of an RccSet object (make sure to set the log.transform parameter accordingly)
- **log.transform** Scalar boolean
- **pdata** pData() of the RccSet object
- **covar** character; colname in pData() that can be used to label genes by a category of interest (passed on to colByCovar)
- **...** additional plotting parameters
Value
A density plot

Author(s)
Dorothee Nickles

Description
Example data for the NanoStringQCPro package

Format
An RccSet object

Author(s)
Dorothee Nickles

Source
This is an artificial dataset designed to resemble real data.

flagSamplesCount, RccSet-method

flagSamplesCount

Description
Flag samples based on overall counts

Usage
## S4 method for signature 'RccSet'
flagSamplesCount(rccSet, method = c("cutoffByMMAD", 
"cutoffByVar"), stringency = 4, maxMiss = 0.2)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rccSet</td>
<td>An RccSet object</td>
</tr>
<tr>
<td>method</td>
<td>Character string specifying the method for outlier detection: either &quot;cutoffByMMAD&quot; or &quot;cutoffByVar&quot;</td>
</tr>
<tr>
<td>stringency</td>
<td>Numeric value passed to the cutoff function specified by the method argument (see the 'd' argument of cutoffByMMAD and cutoffByVar).</td>
</tr>
<tr>
<td>maxMiss</td>
<td>Numeric specifying the allowable fraction of genes below the lower limit of detection in a sample.</td>
</tr>
</tbody>
</table>
Details

The method and stringency arguments determine a cutoff value used to flag samples as outliers: samples will be flagged if the sum of counts of their endogeneous genes exceeds the cutoff or if the ratio of the sums of their positive controls to the sums of their endogenous genes exceeds three times the cutoff. Samples will also be flagged if the fraction of genes below the lower limit of detection exceeds the maxMiss value.

Value

A numeric vector giving the indices of samples with outlier values according to the criteria described above.

Author(s)

Dorothee Nickles
Flag samples based on their technical performance, i.e. field of vision (FOV) counted and binding density.

**Usage**

```r
## S4 method for signature 'RccSet'
flagSamplesTech(rccSet)
```

**Arguments**

- `rccSet` An RccSet object

**Details**

Samples with a FOV counted/FOV count of less than 80

**Value**

A numeric vector giving the indices of samples with outlier values in FOV counted and binding density < 0.05 or > 2.25 (thresholds defined by NanoString) will be flagged.

**Author(s)**

Dorothee Nickles

---

Fields of view (FOV) plot

**Description**

Plot the fraction of successfully imaged fields of view (FOV) in the given RccSet. The RccSet’s phenoData should have ‘FovCount’ and ‘FovCounted’ columns populated with the total and successfully imaged FOV counts, respectively. Samples with a FOV counted/FOV count of less than 80

**Usage**

```r
## S4 method for signature 'RccSet'
fovPlot(rccSet)
```

**Arguments**

- `rccSet` An RccSet object
getBackground,RccSet-method

Value
A plot

Author(s)
Dorothee Nickles

Description
Gene clustering heatmap

Usage
geneClustering(rccSet, outputFile, main = "Gene clustering", covar = NULL)

Arguments
rccSet An RccSet
outputFile Output PNG filename
main Plot title
covar Colname in the rccSet’s fData that can be used to label genes by a category of interest

Value
A PNG file showing clustering of genes by correlation across an experiment Positive and negative control probes and any zero-variance genes (typically housekeeping genes) are omitted from the heatmap. The width and height of the PNG file are set to vary with the size of the input.

Author(s)
Dorothee Nickles, Robert Ziman

getBackground,RccSet-method
Get background estimates for a NanoString RccSet

Description
Returns background estimates for a NanoString RccSet object. The function depends upon correct annotation in the RccSet: if the bgReference argument is set to "blanks", it expects blank measurements (i.e., water runs) to have their phenoData SampleType set to the value indicating blanks (see getBlankLabel()); normally this value would have been set using an argument to newRccSet()). If bgReference is set to "negatives", then it expects to find the negative control probes via CodeClass == "Negative". If set to "both", it expects both of the above and will calculate initial background estimates using an algorithm that mimics the implementation in NanoString’s nSolver Analysis Software (see the nSolverBackground() man page for details on the algorithm).
Usage

## S4 method for signature 'RccSet'
getBackground(rccSet, bgReference = c("both", "blanks", "negatives"), summaryFunction = "median", stringency = 0, nSolverBackground.shrink = TRUE, nSolverBackground.w1 = 2.18, inputMatrix = c("posCtrlData", "exprs"))

Arguments

- **rccSet**: NanoString RccSet object.
- **bgReference**: Measurements to use for background estimates: one of "blanks" (for blank samples), "negatives" (for negative control probes), or "both". Blanks are assumed to be indicated as in the description above.
- **summaryFunction**: Summary function for background measurements (e.g. "mean" or "median"). User-defined functions similar to these can be specified here as well.
- **stringency**: Factor by which deviation (SD or MAD) of the summarization output will be multiplied to obtain final background estimates.
- **nSolverBackground.shrink**: Value to use for the ‘shrink’ argument to nSolverBackground().
- **nSolverBackground.w1**: Value to use for the ‘w1’ argument to nSolverBackground().
- **inputMatrix**: Name of the matrix in the RccSet’s assayData to use as input for calculating background estimates (one of "exprs" or "posCtrlData"). If posCtrlData is specified but not present in the assayData, an error will be generated.

Value

A matrix containing background estimates for a NanoString RccSet object.

Author(s)

Dorothee Nickles

See Also

subtractBackground

Examples

data(example_rccSet)

## Calculate probe-specific background based on negative control probes
bg <- getBackground(example_rccSet, bgReference="negatives", summaryFunction="mean", inputMatrix="exprs")

## Calculate sample-specific background based on blanks
bg <- getBackground(example_rccSet, bgReference="blanks", inputMatrix="exprs")

## Calculate background that is both sample- and probe-specific
bg <- getBackground(example_rccSet, bgReference="both", stringency=1, inputMatrix="exprs")
getBlankLabel,RccSet-method

*Get the SampleType value that indicates blank samples*

**Description**

Returns the phenoData SampleType value that indicates blank samples (i.e., water runs). This value is parsed from the single-quoted string enclosed by "blankLabel='...'") in the varMetadata for SampleType.

**Usage**

```r
## S4 method for signature 'RccSet'
getBlankLabel(rccSet, showWarnings = TRUE)
```

**Arguments**

- `rccSet`: An RccSet
- `showWarnings`: Logical. If FALSE, no warnings will be generated (if any).

**Value**

NULL if the SampleType column is missing altogether, NA if the varMetadata doesn’t have blankLabel recorded, or the blankLabel value otherwise.

**Author(s)**

Robert Ziman

**Examples**

```r
data(example_rccSet)
blankLabel <- getBlankLabel(example_rccSet)
```

---

**getSpikeInInput**

**Description**

Gets the RNA “spike-in” input levels for positive and negative control probes from the label in their GeneName. Note that this is a helper function for readRlf() and elsewhere and is not intended for external use.

**Usage**

```r
g getSpikeInInput(CodeClass, GeneName, nonCtrlProbeVal = NA)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeClass</td>
<td>Character vector with code classes for each probe.</td>
</tr>
<tr>
<td>GeneName</td>
<td>Character vector with gene names/symbols for each probe.</td>
</tr>
<tr>
<td>nonCtrlProbeVal</td>
<td>Value to assign as the spike-in input for the non-control probes.</td>
</tr>
</tbody>
</table>

Value

A data frame with the input CodeClass and GeneName but where the latter has been split into two columns: one showing the GeneName for each probe with spike-in input labels removed – and another with the spike-in input levels.

Author(s)

Robert Ziman

Description

Plot the interquartile range (IQR) for a certain code class of probes in an RccSet object.

Usage

```r
## S4 method for signature 'RccSet'
iqrPlot(rccSet, codeClass = c("Negative", "Positive", "Endogenous", "Housekeeping"), method = c("cutoffByMMAD", "cutoffByVar"), stringency = 4)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rccSet</td>
<td>An RccSet object</td>
</tr>
<tr>
<td>codeClass</td>
<td>Character string specifying the code class (as annotated in the fData(rccSet)$CodeClass column) for which the IQR shall be determined.</td>
</tr>
<tr>
<td>method</td>
<td>Character string specifying the method for outlier detection: either &quot;cutoffByMMAD&quot; or &quot;cutoffByVar&quot;.</td>
</tr>
<tr>
<td>stringency</td>
<td>Numeric value passed to the cutoff function specified by the method argument (see the ‘d’ argument of cutoffByMMAD and cutoffByVar).</td>
</tr>
</tbody>
</table>

Details

IQR of the specified code class for each sample in the RccSet are plotted and outliers (as determined by the function specified in the method argument) are marked in red (thresholds for outlier definition are plotted as red dashed lines).

Value

A plot
Author(s)

Dorothee Nickles

See Also

cutoffByMMAD, cutoffByVar

lodAssess, RccSet-method

lodAssess

Description

Assess how many genes in each sample in an RccSet object are below the limit of detection. (The current implementation does a straightforward column sum on the presence/absence matrix (paData) in assayData.)

Usage

## S4 method for signature 'RccSet'
lodAssess(rccSet)

Arguments

rccSet An RccSet object

Value

A numeric vector giving the number of missing genes (endogenous and housekeeping genes) for each sample in an RccSet. If paData is not found in the input's assayData, NULL is returned.

Author(s)

Dorothee Nickles

lodPlot, RccSet-method

lodPlot

Description

Function to plot the number of missing genes per sample in an RccSet object.

Usage

## S4 method for signature 'RccSet'
lodPlot(rccSet, maxMiss = 0.2)
**Arguments**

- **rccSet** An RccSet object.
- **maxMiss** Numeric specifying the allowable fraction of genes below the lower limit of detection in a sample.

**Details**

Samples with more than 50 measurements are present, they are represented as triangles.

**Value**

A plot

**Author(s)**

Dorothee Nickles

---

**Description**

Creates an html QC report for an RccSet object. Alongside the html file, a directory with matching filename is produced that contains additional files as well as high resolution versions of the various plots in the report. In addition to generating the QC report, the function returns a copy of the input RccSet with columns added to phenoData that show the QC flags for each sample.

**Usage**

```r
## S4 method for signature 'RccSet'
makeQCReport(rccSet, 
  outputBaseName = "NanoStringQCPro_QC_report", outputDir = getwd(), 
  preprocOverride = FALSE, 
  experimentTitle = expinfo(experimentData(rccSet))[["title"], 
  covar = "SampleType", method = c("cutoffByMMAD", "cutoffByVar"), 
  stringency = 4, maxMiss = 0.2, sampleNameCol = "SampleID", 
  heatmaps = FALSE, cleanMarkdown = TRUE, verbose = FALSE)
```

**Arguments**

- **rccSet** RccSet object for which to generate the QC report.
- **outputBaseName** Character string specifying the base filename (without extension) to use for the output file.
- **outputDir** Character string specifying the path to the output directory for the QC report and associated files.
- **preprocOverride** Logical. If TRUE, the input’s preprocessing will be ignored, and a default preprocessing configuration (specifically, the defaults for preprocRccSet()) will be applied so that all applicable plots can be rendered in the report.
myCols

**experimentTitle**
Character string specifying an easy to read identifier of the experiment.

**covar**
Character string specifying a covariate for stratifying samples (e.g. "Sample-Type").

**method**
Method to determine outlier samples: either "cutoffByVar" or "cutoffByMMAD".

**stringency**
Multiplier with which to adjust cutoff values for determining outlier samples.

**maxMiss**
Numeric specifying the allowable fraction of genes below the lower limit of detection in a sample.

**sampleNameCol**
Character string specifying the name of the phenoData column holding the sample names.

**heatmaps**
Logical: render and show heatmaps?

**cleanMarkdown**
Logical: upon completion, delete markdown files used to produce QC report?

**verbose**
Logical: print progress messages?

**Value**
An html report is written to disk and a copy of the input RccSet is invisibly returned with columns added to phenoData that show the QC flags for each sample.

**Author(s)**
Dorothee Nickles, Thomas Sandmann, Robert Ziman, Richard Bourgon

**Examples**
```r
data(example_rccSet)
norm_example_rccSet <- preprocRccSet(example_rccSet)
qc_example_rccSet <- makeQCReport(norm_example_rccSet, "example_QC_report")
```

---

**Description**
Function that defines nice colors

**Usage**
myCols()

**Value**
A vector of colors

**Author(s)**
Dorothee Nickles
Description

NanoStringQCPro

negCtrlsByLane,RccSet-method

negCtrlsByLane

Description

Plot negative controls per lane in an RccSet object

Usage

```r
## S4 method for signature 'RccSet'
negCtrlsByLane(rccSet)
```

Arguments

- `rccSet` : An RccSet object

Details

Boxplots are colored by lane (as specified in the pData slot). Bars on top of the panel indicate the stage position for each cartridge/sample (as specified in the pData slot).

Value

A plot with boxplots for the negative control counts for each individual sample (lane-specific background)

Author(s)

Dorothee Nickles
Description

Plot of negative controls by lane (vertical orientation)

Usage

```r
negCtrlsByLane_verticalPlot(rccSet, outputFile)
```

Arguments

- `rccSet`: An RccSet
- `outputFile`: Output PNG filename

Value

A PNG file containing a boxplot of the counts for negative controls by lane in the input. The PNG is set to a fixed resolution of 300 pixels per inch and a fixed width of 2250 pixels (i.e. 7.5" at 300ppi), but the height varies with the size of the input. The font size is also fixed so that the labels will be legible even for large datasets.

Description

Pairs plot of negative controls across all samples in an RccSet object

Usage

```r
## S4 method for signature 'RccSet'
negCtrlsPairs(rccSet, log.transform = FALSE)
```

Arguments

- `rccSet`: An RccSet object
- `log.transform`: boolean, whether data needs to be log2 transformed

Value

Pairs plot of the negative controls with a scatter plot in the lower panel and correlation coefficients printed in the upper panel.

Author(s)

Dorothee Nickles
Description

Plot negative controls across all samples in an RccSet object

Usage

## S4 method for signature 'RccSet'
negCtrlsPlot(rccSet)

Arguments

rccSet  
An RccSet object

Details

In the second panel, boxplots are colored by lane (as specified in the pData slot). Bars on top of the panel indicate the stage position for each cartridge/sample (as specified in the pData slot).

Value

A plot with two panels: one showing boxplots for the individual negative controls across all samples, and one showing boxplots for the negative control counts for each individual sample (lane-specific background).

Author(s)

Dorothee Nickles

---

newRccSet  
Create a new RccSet object

Description

This is the main wrapper function for generating an RccSet from NanoString data. The function takes as input a vector of NanoString .RCC files with the raw data or a .CSV file generated via the RCC Collector Tool Export feature of NanoString’s nSolver Analysis Software, an optional path to the .RLF file describing the codeset used, optional paths to additional annotation about the features and samples, and details about the experiment. It returns an RccSet object.
Usage


Arguments

rccFiles Vector of paths to .RCC files with the raw count data.
rccCollectorToolExport Path to a .CSV file generated via the RCC Collector Tool Export feature of NanoString's nSolver Analysis Software. (Note that this is an alternative to rccFiles, and if both arguments are specified at the same time, the function will throw an error.)
rlf Path to the NanoString .RLF file describing the codeset used in generating the .RCCs.
cdrDesignData Path to a .CSV extract of the "Design Data" tab of a CDR spreadsheet corresponding to the rest of the input files. See 'Details' section of the buildCodersetAnnotation() help page for more info on how this extract should be prepared.
extraPdata Vector of paths to files containing additional annotation about the samples which will be added to the phenoData of the output RccSet. All files should be tab-separated and should contain a column labelled "FileName" whose values correspond exactly to the basenames (including .RCC extension) of the files specified in rccFiles or listed in the RCC Collector Tool Export. More than one such file may be used. A SampleType column should be present in at most one file.
blankLabel Value for the output’s phenoData SampleType column that will indicate blank samples. This will be recorded in the varMetadata for SampleType. Blank samples, if available, play an important role in preprocessing.
addEgAnnotations Logical indicating whether or not to add EntrezGene annotations from the org.Hs.eg.db package.
dropPdataCols Character vector specifying phenoData columns to be dropped from the output object (if empty or NULL, no columns will be dropped).
dropFdataCols Character vector specifying featureData columns to be dropped from the output object (if empty or NULL, no columns will be dropped).
experimentData.name String passed to the 'name' slot of the output RccSet’s experimentData.
experimentData.lab String passed to the 'lab' slot of the output RccSet’s experimentData.
experimentData.contact String passed to the 'contact' slot of the output RccSet’s experimentData.
experimentData.title String passed to the 'title' slot of the output RccSet’s experimentData.
nSolverBackground,RccSet-method

experimentData.abstract
String passed to the 'abstract' slot of the output RccSet's experimentData.

experimentData.url
String passed to the 'url' slot of the output RccSet's experimentData.

experimentData.other
List passed to the 'other' slot of the output RccSet's experimentData.

Details

In the .RLF (and sometimes in the .RCC files), the GeneName field for positive and negative control probes contains a parenthesized label indicating the RNA "spike-in" levels for each probe. These labels are removed from the control probe GeneNames in the output and recorded instead in SpikeInInput in the output's featureData.

A pseudocount of 1 is added to all measurements to enable subsequent log transformation of the data.

If the phenoData SampleType column is not specified via an annotation file passed in through extraPdata, it will be created and assigned NA for all samples.

Value

An RccSet containing the raw NanoString data and annotations.

Author(s)

Robert Ziman

Examples

```r
rccDir <- system.file("extdata", "RCC", package="NanoStringQCPro")
rccSet <- newRccSet(
  rccFiles = dir(rccDir, full.names=TRUE),
  rlf = system.file("extdata", "RLF", "NQCP_example.rlf", package="NanoStringQCPro"),
  extraPdata = system.file("extdata", "extraPdata", "SampleType.txt", package="NanoStringQCPro"),
  blankLabel = "blank",
  experimentData.name = "Robert Ziman",
  experimentData.lab = "Richard Bourgon",
  experimentData.contact = "ziman.robert@gene.com",
  experimentData.title = "NanoStringQCPro example dataset",
  experimentData.abstract = "Example data for the NanoStringQCPro package"
)
```

nSolverBackground,RccSet-method

nSolver Analysis Software background estimation

Description

Calculates initial probe- and lane-specific background estimates using an algorithm that mimics the implementation in NanoString's nSolver Analysis Software (see details below for the exact algorithm).
Usage

```r
## S4 method for signature 'RccSet'
nSolverBackground(rccSet, stringency = 1, shrink = TRUE,
                 w1 = 2.18, inputMatrix = c("posCtrlData", "exprs"))
```

Arguments

- `rccSet`: NanoString RccSet object
- `stringency`: Multiplier with which to adjust final values.
- `shrink`: Boolean specifying if probe-specific estimates should be shrunken towards their global mean.
- `w1`: Shrink weight "w1".
- `inputMatrix`: Name of the matrix in the RccSet's assayData to use as input for calculating background estimates (one of "exprs" or "posCtrlData"). If posCtrlData is specified but not present in the assayData, an error will be generated.

Details

The mean values for each blank lane (not including positive control probes) are computed from the original data, and a vector of probe-specific background is established by taking the rowMeans of the blank measurements for each probe after subtracting out these values. If `shrink=TRUE`, the vector is adjusted via the following formula (where 'probe.bg' represents the vector):

```r
w2 <- 1/length(blanks)
probe.bg <- (w1*probe.bg + w2*mean(probe.bg)) / (w1 + w2)
```

This probe-specific background is further adjusted by subtracting the mean of its values for the negative control probes. A lane-specific “affinity” is calculated for all lanes in the original data by taking the colMeans of the negative control probe values in the original data, and background estimates for each probe and lane in the original data are computed by summing the corresponding probe-specific background and lane-specific affinity. Any resulting values less than zero are set to zero, and the last step before returning these values is to multiply them by the given stringency.

Value

A matrix containing lane- and probe-specific background estimates.

Author(s)

Dorothee Nickles, Thomas Sandmann

See Also

`getBackground`, `subtractBackground`
**nSolverCsv.to.pdata_fdata_adata**

**Description**
First stage of readRccCollectorToolExport(): produces a list containing matrices (for pdata and adata) and a data frame (for fdata) that pdata_fdata_adata.to.rccSet then transforms into a full RccSet (after some further checks and adjustments). Not intended for external use; see also rccFiles.to.pdata_fdata_adata().

**Usage**
nSolverCsv.to.pdata_fdata_adata(rccCollectorToolExport)

**Arguments**
- rccCollectorToolExport
  Path to the nSolver RCC Collector Tool .CSV export.

**Value**
A list containing matrices (for pdata and adata) and a data frame (for fdata) that pdata_fdata_adata.to.rccSet() then transforms into a full ExpessionSet.

**Author(s)**
Dorothee Nickles, Thomas Sandmann, Robert Ziman

---

**panelCor**

**Description**
Helper function for printing correlation coefficients inside a pairs plots

**Usage**
panelCor(x, y, digits = 2, cex.cor = 0.75, doTest = FALSE)

**Arguments**
- x
  integer
- y
  integer, same length as x
- digits
  scalar integer, indicating the number of decimal positions for displaying the correlation coefficient
- cex.cor
  scalar numeric to specify relative font size for printing the correlation coefficient
- doTest
  boolean, whether a results of cor.test should be displayed as well
pcaPlot

Description
Wrapper function to perform a PCA analysis on the exprs slot of an RccSet object and plot some results.

Usage
pcaPlot(exx, ...)

Arguments
- **exx**: exprs() of an RccSet object
- ...
  additional parameters passed on to the plotting functions

Value
PCA screeplot and a plot with two panels, one plotting PC1 versus PC2, the other plotting PC1 versus PC3.

Author(s)
Dorothee Nickles

pdata_fdata_adata.to.rccSet

Description
Second stage of readRccBatch()/readRccCollectorToolExport() – not intended for external use.

Usage
pdata_fdata_adata.to.rccSet(pdata_fdata_adata)

Arguments
- **pdata_fdata_adata**: List containing the pdata, fdata, and adata returned by rccFiles.to.pdata_fdata_adata() or nSolverCsv.to.pdata_fdata_adata().

Value
Prints correlation coefficients (and p-values if doTest = TRUE) within a pairs plot.
Details

Note that a pseudo-count of 1 is always added to all measurements, to enable subsequent log transformation of the data in cases where zero-counts are present.

N.B. The function currently expects certain columns to be present in pdata_fdata_a_data$pdata.m, and it converts these to numerics. These expectations should be incorporated into the class definition, and conversion should only take place with a warning. Future updates will address this.

Value

An RccSet whose contents reflect the input data.

Author(s)

Robert Ziman

Description

Applies positive control normalization to the data in an RccSet object.

Usage

## S4 method for signature 'RccSet'
posCtrlNorm(rccSet, summaryFunction = "sum", quietly = FALSE)

Arguments

rccSet
An RccSet object.

summaryFunction
Function to be used for the normalization (e.g. "mean", "median", or "sum"). User-defined functions similar to these can be specified here as well.

quietly
Logical. If TRUE, messages and warnings will not be shown.

Value

A copy of the input RccSet that has count data adjusted by positive control counts. The positive control scaling factor is recorded in PosFactor in the output’s phenoData (if this column already exists in the input, it will be overwritten in the output copy).

Author(s)

Dorothee Nickles

Examples

data(example_rccSet)
pcnorm_example_rccSet <- posCtrlNorm(example_rccSet)
posNormFactPlot, RccSet-method

Description
Plot positive control scaling factor for each sample in an RccSet object.

Usage
```r
## S4 method for signature 'RccSet'
posNormFactPlot(rccSet)
```

Arguments
- `rccSet`: An RccSet object

Value
A plot of the positive control scaling factor for each sample in an RccSet object. Samples with a positive control scaling factor < 0.3 or > 3 (thresholds defined by NanoString) are marked in red (dashed red line indicates threshold).

Author(s)
Dorothee Nickles

posR2Plot, RccSet-method

Description
Plot the R squared of linear fit of counts versus input for positive controls in an RccSet object.

Usage
```r
## S4 method for signature 'RccSet'
posR2Plot(rccSet)
```

Arguments
- `rccSet`: RccSet object

Details
R squared for each sample in the RccSet are plotted and samples with R squared < 0.95 are marked in red (threshold indicated by dashed red line).
Value

A plot

Author(s)

Dorothee Nickles

Description

Plot the ratio of the mean of positive control counts for each sample and the overall mean of positive control counts in an RccSet object.

Usage

```r
## S4 method for signature 'RccSet'
posRatioPlot(rccSet, method = c("cutoffByMMAD", "cutoffByVar"), stringency = 4)
```

Arguments

- `rccSet`: An RccSet object
- `method`: Character string specifying the method for outlier detection: either "cutoffByMMAD" or "cutoffByVar".
- `stringency`: Numeric value passed to the cutoff function specified by the method argument (see the ‘d’ argument of cutoffByMMAD and cutoffByVar).

Details

The ratio for each sample in the RccSet is plotted and and outliers (as determined the cutoff function specified by the method argument) are marked in red (thresholds for outlier definition are plotted as red dashed lines).

Value

A plot

Author(s)

Dorothee Nickles
posSlopePlot,RccSet-method

Description

Plot the slope of linear fit of counts versus input for positive controls in an RccSet object.

Usage

```r
## S4 method for signature 'RccSet'
posSlopePlot(rccSet, method = c("cutoffByMMAD", "cutoffByVar"), stringency = 4)
```

Arguments

- `rccSet`: An RccSet object
- `method`: Character string specifying the method for outlier detection: either "cutoffByMMAD" or "cutoffByVar".
- `stringency`: Numeric value passed to the cutoff function specified by the method argument (see the ‘d’ argument of cutoffByMMAD and cutoffByVar).

Details

The slope for each sample in the RccSet are plotted and and outliers (as determined by the function specified by the method argument) are marked in red (thresholds for outlier definition are plotted as red dashed lines).

Value

A plot

Author(s)

Dorothee Nickles

posSumVsAllSumPlot,RccSet-method

Description

Plot the ratio of sums of positive control counts to all counts for all samples in an RccSet object.

Usage

```r
## S4 method for signature 'RccSet'
posSumVsAllSumPlot(rccSet, method = c("cutoffByMMAD", "cutoffByVar"), stringency = 4)
```
Arguments

rccSet  An RccSet object
method  Character string specifying the method for outlier detection: either "cutoffByM-MAD" or "cutoffByVar".
stringency  Numeric value passed to the cutoff function specified by the method argument (see the ‘d’ argument of cutoffByMMAD and cutoffByVar). (If the median ratio is less than 1, three times this value will be used.)

Details

The ratio for each sample in the RccSet is plotted and and outliers (as determined by the cutoff function specified by the method argument) are marked in red (thresholds for outlier definition are plotted as red dashed lines).

Value

A plot

Author(s)

Dorothee Nickles

preprocRccSet,RccSet-method

Preprocess an RccSet

Description

This function is a wrapper to perform any combination of positive control normalization, background correction, and content normalization on the input RccSet. For each completed preprocessing step, a matrix is added to the assayData of the resulting RccSet object:

- posCtrlData: expression data after positive control normalization
- bgEstimates: background estimates
- bgCorrData: expression data after positive control normalization and background correction
- normData: expression data after positive control normalization, background correction, and content normalization

(NOTE: normData is on a log2 scale while all the other matrices are on a linear scale.)

If any step is omitted, the corresponding matrix will not be present in the output’s assayData. The parameters for all steps are recorded in the output’s experimentData@preprocessing list (accessible through preproc(rccSet) where rccSet is an RccSet output by this function). In addition:

- If positive control normalization is performed, a column named 'PosCtrl' is added to the output’s phenoData to record the positive control scaling factors.
- If the presence/absence call is performed, a matrix named ‘paData’ is added to the output’s assayData to indicate the presence/absence of each feature in each sample. See the ‘pa’ argument for details.
- If housekeeping normalization is performed, a column labeled ‘Housekeeping’ is added to the featureData to indicate which features were used for it.
Usage

```r
## S4 method for signature 'RccSet'
preprocRccSet(rccSet, doPosCtrlNorm = TRUE,
               doBackground = TRUE, doPresAbs = TRUE, doContentNorm = TRUE,
               pcnSummaryFunction = "sum", bgReference = c("both", "blanks", "negatives"),
               bgSummaryFunction = "median", bgStringency = 1,
               nSolverBackground.w1 = 2.18, nSolverBackground.shrink = TRUE,
               paStringency = 2, normMethod = c("global", "housekeeping"),
               normSummaryFunction = "median", hkgenes = NULL, hkfeatures = NULL,
               quietly = FALSE)
```

Arguments

- `rccSet`: An RccSet.
- `doPosCtrlNorm`: Boolean specifying whether or not to perform positive control normalization. (`pcd` is short for `posCtrlData`, the matrix which gets added to assayData when this step is performed.)
- `doBackground`: Boolean specifying whether or not to perform background correction.
- `doPresAbs`: Boolean specifying whether or not the presence/absence call should be performed. For details, see presAbsCall().
- `doContentNorm`: Boolean specifying whether or not content normalization should be performed.
- `pcnSummaryFunction`: Function to be used for the positive control normalization (e.g. "mean", "median", or "sum"). User-defined functions similar to these can be specified here as well.
- `bgReference`: Measurements to use for background estimates: either "blank" (for blank samples), "negatives" (for negative control probes), or "both". For details on exactly how the background estimates are computed in each case, see getBackground().
- `bgSummaryFunction`: Summary function for background measurements (e.g. "mean" or "median"). User-defined functions similar to these can be specified here as well.
- `bgStringency`: Factor by which deviation (SD or MAD) of the summarization output will be multiplied to obtain final background estimates.
- `nSolverBackground.w1`: Value to use for the ‘w1’ argument to nSolverBackground(). (Only takes effect if bgReference == "both"; see getBackground().)
- `nSolverBackground.shrink`: Value to use for the ‘shrink’ argument to nSolverBackground(). (Only takes effect if bgReference == "both"; see getBackground().)
- `paStringency`: Multiplier to use in establishing the presence/absence call. For details, see presAbsCall().
- `normMethod`: Specifies the features to be used for content normalization. "global" indicates that all features should be used and "housekeeping" indicates that only housekeeping features should be used. If "housekeeping" is specified and the 'hk' argument (below) is also specified, then the features indicated by 'hk' will be used. If "housekeeping" is specified and 'hk' is left NULL, then the default housekeeping features (i.e. those with CodeClass == "Housekeeping") will be used.
- `normSummaryFunction`: Summary function to use for content normalization (e.g. "mean" or "median"). User-defined functions similar to these can be specified here as well.
- `hkgenes`: A vector of gene names to use for content normalization.
- `hkfeatures`: A vector of feature names to use for content normalization.
normSummaryFunction
Character specifying the summary function to apply to the selected features (e.g. "mean" or "median") during the content normalization step. User-defined functions similar to these can be specified here as well.

hkgenes
Character vector with gene symbols to be used for content normalization if housekeeping is specified as the normalization method. If specified, all features that match any of the specified symbols will be used. (To specify specific features, use the 'hkfeatures' argument instead; see below.)

hkfeatures
Character vector with full feature names ("<CodeClass>_<GeneName>_<Accession>", e.g. "Endogenous_ACTG1_NM_001614.1") to be used for content normalization if housekeeping is specified as the normalization method. (Note: if this argument is specified at the same time as 'hkgenes', an error will be thrown.)

quietly
Boolean specifying whether or not messages and warnings should be omitted.

Details
For more information on the rationale behind the recommended preprocessing and normalization steps, please see the vignette.

Value
A copy of the input RccSet with additional matrices in the assayData for each successive preprocessing step along with parameters for each step recorded in the experimentData@preprocessing list.

Author(s)
Dorothee Nickles, Robert Ziman

References
NanoString nCounter(R) Expression Data Analysis Guide (2012)

Examples
```r
data(example_rccSet)
hknorm_example_rccSet <- preprocRccSet(example_rccSet)
```

Description
Adds a matrix to assayData (‘paData’) which indicates the presence/absence call for each gene in each sample using the background estimates and a stringency value. A gene is considered present in a sample if its count in that sample exceeds the corresponding background estimate times the stringency. The count values can be taken from either the positive control normalized data or the raw data (see the inputMatrix argument). If the input doesn’t contain background-corrected data, an error will be generated.
Usage

```r
## S4 method for signature 'RccSet'
presAbsCall(rccSet, stringency = 2,
            inputMatrix = c("posCtrlData", "exprs"), quietly = FALSE)
```

Arguments

- `rccSet`: An `RccSet` with background-corrected data.
- `stringency`: Multiplier to use in establishing the presence/absence call as mentioned in the description.
- `inputMatrix`: Name of the matrix in the `RccSet`'s assayData on which to apply the presence/absence call (either "posCtrlData" or "exprs").
- `quietly`: Logical. If TRUE, messages and warnings will not be shown.

Value

A copy of the input is returned with a new matrix named ‘paData’ added to the assayData that contains the presence/absence calls.

Examples

```r
data(example_rccSet)
pcnorm_rccSet <- posCtrlNorm(example_rccSet)
bgEst <- getBackground(pcnorm_rccSet)
bgcorr_rccSet <- subtractBackground(pcnorm_rccSet, bgEst)
pa_rccset <- presAbsCall(bgcorr_rccSet)
```

---

**previewPNG**

Create a preview of a PNG

Description

Generates a resized, vertically-cropped preview version of the input PNG.

Usage

```r
previewPNG(inputFile, outputFile, width, cropHeight, res = 72)
```

Arguments

- `inputFile`: Input PNG filename
- `outputFile`: Output PNG filename
- `width`: Width (in pixels) for the preview image
- `cropHeight`: Height (in pixels) for the preview image (if the rescaled input is larger than this, it will be cropped)
- `res`: Output PNG resolution (passed to the ‘res’ argument of `png()`)

Value

A resized, vertically-cropped preview version of the input PNG.
Description

First stage of readRccBatch(): produces a list containing matrices (for pdata and adata) and a data frame (for fdata) that pdata_fdata_adata.to.rccSet() then transforms into a full RccSet (after some further checks and adjustments). See also nSolverCsv.to.pdata_fdata_adata().

Usage

rccFiles.to.pdata_fdata_adata(rccFiles)

Arguments

rccFiles  Vector of .RCC paths

Value

A list containing matrices (for pdata and adata) and a data frame (for fdata) that pdata_fdata_adata.to.rccSet() then transforms into a full ExpessionSet.

Author(s)

Robert Ziman

Description

Constructor methods for making new RccSet objects.

Usage

RccSet(obj, ...)

## S4 method for signature 'ExpressionSet'
RccSet(obj, ...)

## S4 method for signature 'environment'
RccSet(obj, ...)

## S4 method for signature 'matrix'
RccSet(obj, ...)

## S4 method for signature 'missing'
RccSet(obj, ...)

RccSet-class

Arguments

obj An object of appropriate class
... Passed to methods.

Details

Arguments accepted by constructors are identical to those for the ExpressionSet constructors. See RccSet class documentation for examples of constructor use. Constructor calls for which mandatory phenoData or featureData columns are missing will successfully create objects that include mandatory columns, but with NA values. See RccSet documentation for a list of mandatory columns.

Value

A new RccSet object.

RccSet-class  RccSet class, derived from ExpressionSet

Description

The RccSet class is a trivial extension of ExpressionSet, but with additional validation criteria. RccSet is a class generator function.

Details

A valid RccSet object must have the following columns in featureData: "CodeClass", "GeneName", and "Accession". It must also have the following phenoData columns: "FileName", "SampleID", "LaneID", "FovCount", "FovCounted", "StagePosition", "BindingDensity", "CartridgeID", and "SampleType". A final requirement is that the "FovCount" column of phenoData have at most one distinct value.

See Also

See checkRccSet, which provides additional checks and generates warnings for unexpected or unusual conditions which, though permitted by the class, may indicate data import errors.

Examples

data("example_rccSet")
e <- example_rccSet

# "ExpressionSet" constructor makes a new assayData environment
r1 <- RccSet(e)
validObject(r1)
assayData(e)
assayData(r1)
head(pData(r1))
head(fData(r1))

# For other constructors, if not explicitly supplied, blank phenoData and
# featureData objects are populated with mandatory columns (and NA values).
r2 <- RccSet(assayData(e))
validObject(r2)
head(pData(r2))
head(fData(r2))

r3 <- RccSet(assayData(e), phenoData(e), featureData(e))
identical(pData(r1), pData(r3))
identical(fData(r1), fData(r3))
identical(annotation(r1), annotation(r3)) # We forgot it!
annotation(e)
r3 <- RccSet(assayData(e), phenoData(e), featureData(e), annotation = annotation(e))
identical(annotation(r1), annotation(r3)) # Better
identical(r1, r3) # False, due to assayData environments
assayData(r1)
assayData(r3)

# Matrix constructor is similar
r4 <- RccSet(exprs(e), phenoData(e), featureData(e), annotation = annotation(e))
identical(exprs(r1), exprs(r4))

# Blank object constructor
r0 <- RccSet()
dim(r0)
pData(r0)
fData(r0)

---

**readCdrDesignData**  
*Read .CSV containing CDR 'Design Data' extract*

**Description**

Return a data frame containing the contents of the 'Design Data' tab extracted from a CDR spreadsheet. The extract, a .CSV file, must be manually prepared in advance (see 'details' section in the buildCodesetAnnotation() help page for more info).

**Usage**

```r
readCdrDesignData(cdrDesignData)
```

**Arguments**

- `cdrDesignData`  
  Path to the .CSV file containing the content extracted from the CDR’s 'Design Data' tab

**Value**

A data frame containing the contents of the CDR 'Design Data' tab.

**Author(s)**

Robert Ziman
Examples

```r
path <- system.file("extdata", "CDR", "CDR-DesignData.csv", package="NanoStringQCPro")
cdr <- readCdrDesignData(path)
```

readRcc Batch

Description

Reads the contents of all .RCC files from a given directory into a new RccSet object. Note: this function is not intended for external use. For that, see newRccSet().

Usage

```r
readRccBatch(rccFiles)
```
Arguments

rccFiles Vector of .RCC file paths

Value

An RccSet object that has raw counts in assayData, probe information in fData, and sample annotation in pData.

Author(s)

Robert Ziman

Description

Reads the contents of a .CSV file generated from the RCC Collector Tool Export feature of NanoString’s nSolver Analysis software into a new RccSet object. (Note: this function is not intended for external use. For that, see newRccSet().)

Usage

readRccCollectorToolExport(file)

Arguments

file Path to the NSolver .CSV file to be read.

Details

See `details` in the readRccBatch() help page.

Value

An RccSet object that has count data in exprs, probe information in fData and sample annotation in pData.

Author(s)

Dorothee Nickles, Thomas Sandmann
### readRlf

**Read RLF file**

### Description

Reads the contents of an .RLF file into a data frame. RNA “spike-in” concentrations recorded in the GeneName for positive and negative control probes are stripped and stored in a separate column in the output. An error will be generated for any recognized deviations from the expected file format.

### Usage

```r
readRlf(rlf)
```

### Arguments

- **rlf**
  - Path to the .RLF file

### Value

A data frame containing the contents of the .RLF file.

### Author(s)

Robert Ziman

### Examples

```r
rlf <- system.file("extdata", "RLF", "NQCP_example.rlf", package="NanoStringQCPro")
rlf.df <- readRlf(rlf)
```

---

### sampleClustering.RccSet-method

**Clustering by sample correlation**

### Description

Clustering by sample correlation

### Usage

```r
## S4 method for signature 'RccSet'
sampleClustering(rccSet, outputFile,
    main = "Sample correlations in raw data", annCol = NULL,
    covar = "SampleType")
```
**scatterPair**

**Arguments**

- **rccSet**: An RccSet
- **outputFile**: Output PNG filename
- **main**: Plot title
- **annCol**: See `aheatmap`
- **covar**: Covariate (e.g. "SampleType")

**Value**

A PNG file showing clustering of samples by correlation. Any zero-variance samples are omitted from the heatmap. The width and height of the PNG file are set to vary with the size of the input.

**Author(s)**

Dorothee Nickles, Robert Ziman

---

### scatterPair

**Description**

Helper function for a scatter plot inside a pairs plots

**Usage**

```
scatterPair(x, y)
```

**Arguments**

- **x**: integer, x positions
- **y**: integer, y positions

**Value**

A scatter plot x versus y.
subtractBackground,RccSet-method

Subtract background estimates for a NanoString RccSet

Description

Returns a NanoString RccSet with background-corrected count data. During subtraction, any counts below 1 will be truncated to 1 to enable subsequent log transformation of the data.

Usage

```r
## S4 method for signature 'RccSet'
subtractBackground(rccSet, bgEstimates, 
               bgEstimatesParams = list(), inputMatrix = c("posCtrlData", "exprs"),
               quietly = FALSE)
```

Arguments

- `rccSet` NanoString RccSet object
- `bgEstimates` Matrix containing the background estimates to subtract.
- `bgEstimatesParams` A list with the parameters that were used to generate the background estimates (see `getBackground()`):
  - `bgReference`
  - `summaryFunction`
  - `stringency`
  - `nSolverBackground.w1`
  - `nSolverBackground.shrink`
  - `inputMatrix`

The values of these list elements will be assigned to corresponding elements in the output’s experimentData@preprocessing list. If any element is NULL, the corresponding element in the output’s preprocessing list will be NA.

- `inputMatrix` Name of the matrix in the RccSet’s assayData to use as input for subtracting background estimates (one of "exprs" or "posCtrlData"). If posCtrlData is specified but not found in the assayData, an error will be generated.
- `quietly` Boolean specifying whether or not messages and warnings should be omitted.

Value

A NanoString `linkS4class(RccSet)` object with background estimates subtracted from the count data.

Author(s)

Dorothee Nickles

See Also

getBackground
Examples

data(example_rccSet)

pcnorm_rccSet <- posCtrlNorm(example_rccSet)

bg1 <- getBackground(pcnorm_rccSet, bgReference="negatives", summaryFunction="mean")
bg2 <- getBackground(pcnorm_rccSet, bgReference="blanks")
bg3 <- getBackground(pcnorm_rccSet, bgReference="both", stringency=1)

bgCor1 <- subtractBackground(pcnorm_rccSet, bgEstimates=bg1)
bgCor2 <- subtractBackground(pcnorm_rccSet, bgEstimates=bg2)
bgCor3 <- subtractBackground(pcnorm_rccSet, bgEstimates=bg3)

---

zfacFun

Description

Calculate Z' Factor

Usage

zfacFun(p, n)

Arguments

p numeric vector: measurements for the positive controls (or actual measurement)
n numeric vector: measurements for the negative controls

Value

Scalar numeric: the Z' Factor

Author(s)

Dorothee Nickles
Index

*Topic* **datasets**

example_rccSet, 15
.RccSet (RccSet-class), 42

addCodesetAnnotation
  (addCodesetAnnotation, RccSet-method), 3
addCodesetAnnotation, RccSet-method, 3
addQCFlags (addQCFlags, RccSet-method), 4
addQCFlags, RccSet-method, 4
aheatmap, 47
allSumPlot (allSumPlot, RccSet-method), 4
allSumPlot, RccSet-method, 4
assessHousekeeping
  (assessHousekeeping, RccSet-method), 5
assessHousekeeping, RccSet-method, 5
bdPlot (bdPlot, RccSet-method), 6
bdPlot, RccSet-method, 6
buildCodesetAnnotation, 6
checkRccSet, 42
checkRccSet
  (checkRccSet, RccSet-method), 7
checkRccSet, RccSet-method, 7
colByCovar, 8
colByFun, 9
contentNorm
  (contentNorm, RccSet-method), 9
contentNorm, RccSet-method, 9
copyRccSet (copyRccSet, RccSet-method), 10
copyRccSet, RccSet-method, 10
countsInBlankSamples_verticalPlot, 11
ctrlsOverviewPlot
  ( ctrlsOverviewPlot, RccSet-method), 12
ctrlsOverviewPlot, RccSet-method, 12
ctrlsZprimePlot
  ( ctrlsZprimePlot, RccSet-method), 12
ctrlsZprimePlot, RccSet-method, 12
cutoffByMMAD, 13, 22
cutoffByVar, 13, 22
dCoVar, 14
densityPlot, 14
example_rccSet, 15
ExpressionSet, 42
flagSamplesCount
  (flagSamplesCount, RccSet-method), 15
flagSamplesCount, RccSet-method, 15
flagSamplesCtrl
  (flagSamplesCtrl, RccSet-method), 16
flagSamplesCtrl, RccSet-method, 16
flagSamplesTech
  (flagSamplesTech, RccSet-method), 17
flagSamplesTech, RccSet-method, 17
fovPlot (fovPlot, RccSet-method), 17
fovPlot, RccSet-method, 17
geneClustering, 18
getBackground, 30, 48
getBackground
  (getBackground, RccSet-method), 18
getBackground, RccSet-method, 18
getBlankLabel
  (getBlankLabel, RccSet-method), 20
getBlankLabel, RccSet-method, 20
getSpikeInInput, 20
iqrPlot (iqrPlot, RccSet-method), 21
iqrPlot, RccSet-method, 21
lodAssess (lodAssess, RccSet-method), 22
lodAssess, RccSet-method, 22
lodPlot (lodPlot, RccSet-method), 22
lodPlot, RccSet-method, 22
makeQCRreport
  (makeQCRreport, RccSet-method), 23

50
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>makeQCReport, RccSet-method</td>
<td>23</td>
</tr>
<tr>
<td>myCols</td>
<td>24</td>
</tr>
<tr>
<td>NanoStringQCPro</td>
<td>25</td>
</tr>
<tr>
<td>NanoStringQCPro-package</td>
<td>(NanoStringQCPro), 25</td>
</tr>
<tr>
<td>negCtrllsByLane</td>
<td>(negCtrllsByLane, RccSet-method), 25</td>
</tr>
<tr>
<td>negCtrllsByLane,RccSet-method</td>
<td>25</td>
</tr>
<tr>
<td>negCtrllsByLane,RccSet-method</td>
<td>25</td>
</tr>
<tr>
<td>negCtrllsByLane_verticalPlot</td>
<td>26</td>
</tr>
<tr>
<td>negCtrllsPairs</td>
<td>(negCtrllsPairs, RccSet-method), 26</td>
</tr>
<tr>
<td>negCtrllsPairs,RccSet-method</td>
<td>26</td>
</tr>
<tr>
<td>negCtrllsPlot</td>
<td>(negCtrllsPlot, RccSet-method), 27</td>
</tr>
<tr>
<td>negCtrllsPlot,RccSet-method</td>
<td>27</td>
</tr>
<tr>
<td>newRccSet</td>
<td>27</td>
</tr>
<tr>
<td>nSolverBackground</td>
<td>(nSolverBackground, RccSet-method), 29</td>
</tr>
<tr>
<td>nSolverBackground,RccSet-method</td>
<td>29</td>
</tr>
<tr>
<td>nSolverCsv.to_pdata_fdata_adata</td>
<td>31</td>
</tr>
<tr>
<td>panelCor</td>
<td>31</td>
</tr>
<tr>
<td>pcaPlot</td>
<td>32</td>
</tr>
<tr>
<td>pdata_fdata_adata.to.rccSet</td>
<td>32</td>
</tr>
<tr>
<td>posCtrl1Norm</td>
<td>(posCtrl1Norm, RccSet-method), 33</td>
</tr>
<tr>
<td>posCtrl1Norm,RccSet-method</td>
<td>33</td>
</tr>
<tr>
<td>posNormFactPlot</td>
<td>(posNormFactPlot, RccSet-method), 34</td>
</tr>
<tr>
<td>posNormFactPlot,RccSet-method</td>
<td>34</td>
</tr>
<tr>
<td>posR2Plot</td>
<td>(posR2Plot, RccSet-method), 34</td>
</tr>
<tr>
<td>posR2Plot,RccSet-method</td>
<td>34</td>
</tr>
<tr>
<td>posRatioPlot</td>
<td>(posRatioPlot, RccSet-method), 35</td>
</tr>
<tr>
<td>posRatioPlot,RccSet-method</td>
<td>35</td>
</tr>
<tr>
<td>posSlopePlot</td>
<td>(posSlopePlot, RccSet-method), 36</td>
</tr>
<tr>
<td>posSlopePlot,RccSet-method</td>
<td>36</td>
</tr>
<tr>
<td>posSumVsAllSumPlot</td>
<td>(posSumVsAllSumPlot, RccSet-method), 36</td>
</tr>
<tr>
<td>posSumVsAllSumPlot,RccSet-method</td>
<td>36</td>
</tr>
<tr>
<td>preprocRccSet</td>
<td>(preprocRccSet, RccSet-method), 37</td>
</tr>
<tr>
<td>preprocRccSet,RccSet-method</td>
<td>37</td>
</tr>
<tr>
<td>presAbsCall</td>
<td>(presAbsCall, RccSet-method), 39</td>
</tr>
<tr>
<td>presAbsCall,RccSet-method</td>
<td>39</td>
</tr>
<tr>
<td>previewPNG</td>
<td>40</td>
</tr>
<tr>
<td>rccFiles.to_pdata_fdata_adata</td>
<td>41</td>
</tr>
<tr>
<td>RccSet</td>
<td>29, 33, 41, 41, 42</td>
</tr>
<tr>
<td>RccSet, environment-method</td>
<td>(RccSet), 41</td>
</tr>
<tr>
<td>RccSet, ExpressionSet-method</td>
<td>(RccSet), 41</td>
</tr>
<tr>
<td>RccSet, matrix-method</td>
<td>(RccSet), 41</td>
</tr>
<tr>
<td>RccSet, missing-method</td>
<td>(RccSet), 41</td>
</tr>
<tr>
<td>RccSet-class</td>
<td>42</td>
</tr>
<tr>
<td>readCdrDesignData</td>
<td>43</td>
</tr>
<tr>
<td>readRcc</td>
<td>44</td>
</tr>
<tr>
<td>readRccBatch</td>
<td>44</td>
</tr>
<tr>
<td>readRccCollectorToolExport</td>
<td>45</td>
</tr>
<tr>
<td>readRlf</td>
<td>46</td>
</tr>
<tr>
<td>sampleClustering</td>
<td>(sampleClustering, RccSet-method), 46</td>
</tr>
<tr>
<td>subtractBackground</td>
<td>19, 30</td>
</tr>
<tr>
<td>subtractBackground,RccSet-method</td>
<td>48</td>
</tr>
<tr>
<td>subtractBackground,RccSet-method</td>
<td>48</td>
</tr>
<tr>
<td>zfacFun</td>
<td>49</td>
</tr>
</tbody>
</table>