arrayQuality

October 25, 2011

`MmDEGenes` *Known DE genes for Mouse quality hybridizations.*

Description

MmDEGenes contains information about probes known to be DE from previous quality hybridizations. It is used to verify reproducibility of print-runs.

Source

These data were provided by members of the UCSF Shared functional genomics core lab.

`PRv9mers` *Print Run Quality version 9mers*

Description

Qualitative diagnostic plots looking at print-run quality. This component examine the 9mers hybridizations.

Usage

```
PRv9mers(fnames, path = ".", dev = "png", DEBUG = FALSE, prargs = NULL, samepr =
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only.
`DEBUG`	If 'TRUE', debug statements are printed.
`prargs`	A list with 4 components: Block, Row, Column and ncolumns. See Details for more information.
`samepr`	If 'TRUE', we assume everything in the directory are from the same print-run.
`prname`	A "character" string giving the name of the print-run.
`save`	If 'TRUE', the figures will be saved to files.
`...`	additional arguments

Details

The argument "prargs" is used to calculate the layout information about a print-run. Components Block, Row and Columns denote the column names from the input data representing the print-tip location. The component "ncolumns" is an integer representing the number of print-tip columns in the data. If the argument is set to NULL, the the following default will be used: list(Block="Block", Row="Row", Column="Column", ncolumns=4).

Value

Files of diagnostic plots and excel files containings probe IDs of problematic probes.

Author(s)

Jean Yee Hwa Yang

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
PRv9mers(fnames="12Mm250.gpr", path=datadir, prname="12Mm")
```

`PRvQCHyb` *Print run Quality version Quality Control Hybridization*

Description

Qualitative diagnostic plots looking at print-run quality. This component examine the QC hybridizations.

Usage

```
PRvQCHyb(fnames, path=".", dev = "png", DEBUG=FALSE, prargs=NULL, samepr=TRUE, p
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	a character string representing the data directory. By default this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only.
`DEBUG`	If 'TRUE', debug statements are printed.
`prargs`	A list with 4 components: Block, Row, Column and ncolumns. See Details for more information.
`samepr`	If 'TRUE', we assume everything in the directory are from the same print-run.
`prname`	A "character" string giving the name of the print-run.
`save`	If 'TRUE', the figures will be saved to files.
`col`	color code for different control samples.
`...`	additional arguments.

Details

The argument "prargs" is used to calculate the layout information about a print-run. Components Block, Row and Columns denote the column names from the input data representing the print-tip location. The component "ncolumns" is an integer representing the number of print-tip columns in the data. If the arguement is set to NULL, the the following default will be used. list(Block="Block", Row="Row", Column="Column", ncolumns=4)

Value

Files of diagnostic plots.

Author(s)

Jean Yee Hwa Yang

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
PRvQCHyb(fnames="9Mm137.gpr", path=datadir, prname="9Mm")
```

`agQuality` *Diagnostic plots and comparative boxplots for general hybridization,*

Description

This component provides qualitative diagnostic plots and quantitative measures for assessing general hybridization quality. All results are displayed in a HTML report. Agilent format only.

Usage

```
agQuality(fnames = NULL, path = ".", organism = c("Mm", "Hs"),
compBoxplot = TRUE, reference = NULL, controlMatrix = agcontrolCode,
controlId = c("ProbeName"), output = FALSE, resdir = ".", dev= "png", DEBUG = FA
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`compBoxplot`	Logical. If set to 'FALSE', only qualitative diagnostic plots will be plotted. agQuality ouput will be limited to a diagnostic plot by file and a marrayRaw object.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used. See details for more information.

`controlMatrix`	A character matrix of n by 2 columns. The first column contains a few regular expression of spotted probe sequences and the second column contains the corresponding control status. By default, is it set to be agcontrolCode.
`controlId`	Character string. Name of the column of the file used to define controls.
`output`	Logical. If 'TRUE', normalized M values corresponding to the input Agilent files and quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	Additional arguments.

Details

agQuality returns 2 plots for each Agilent files passed as argument. The first one is a qualitative diagnostic plot, a quick visual way to assess slide quality. The second one is a comparative boxplot: each quality control measure is compared to the range obtained for a database of 'good' slides used as reference. You can use your own set of references created using `globalQuality` passed in the arguments "reference", or use the reference QC values stored in the datasets `MmReferenceDB` and `HsReferenceDB`. All results and quality scores are gathered in a HTML report. For more details about the QC measures and the plots, please refer to the online manual.

Value

A list of 2 elements:

`mraw`	A marrayRaw object created from the input files.
`quality`	A matrix containing Quality Control measures for all slides.

Author(s)

Agnes Paquet

See Also

`globalQuality`, `qualBoxplot`, `readAgilent`

Examples

```
example
```

`Internal functions` *Internal arrayQuality functions*

Description

Internal arrayQuality functions

Details

These are not to be called by the user.

`globalQuality` *Quality measures for general hybridization.*

Description

This function provides Quality Control measures for GenePix, Spot and Agilent format files. It is used to create a table of measures to be used as reference in gpQuality, spotQuality or agQuality.

Usage

```
globalQuality(fnames = NULL, path = ".", organism = c("Mm", "Hs"),
output = FALSE, resdir = ".", DEBUG = FALSE, inputsource = "readGPR", controlId=
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	a "character" string representing the data directory. By default this is set to the current working directory (".").
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`output`	Logical. If 'TRUE', the quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`DEBUG`	If 'TRUE', debug statements are printed.
`inputsource`	A "character" string providing the name of the function to use to read the input files. It should be inputsource = "readGPR" for GenePix format files, inputsource = "readSpot" for Spot files, or inputsource = "readAgilent" for Agilent format. By default, 'inputsource' is set to "readGPR".
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`...`	additional arguments

Value

A matrix of Quality Control measures, each column representing a different input slide.

Author(s)

Agnes Paquet

See Also

`gpQuality`, `slideQuality`, `MmReferenceDB`, `readGPR`, `readSpot`, `readAgilent`

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
reference <- globalQuality(fnames="9Mm137.gpr", path=datadir, organism="Mm")
```

`gpQuality` *Diagnostic plots and comparative boxplots for general hybridization*

Description

This component provides qualitative diagnostic plots and quantitative measures for assessing general hybridization quality. All results are displayed in a HTML report. GenePix format only.

Usage

```
gpQuality(fnames = NULL, path = ".", organism = c("Mm", "Hs"),
compBoxplot = TRUE, reference = NULL, controlMatrix = controlCode,
controlId = c("ID", "Name"), output = FALSE, resdir = ".", dev= "png",
val=c("maM", "maA"),DEBUG = FALSE,...)
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`compBoxplot`	Logical. If set to 'FALSE', only qualitative diagnostic plots will be plotted. gpQuality ouput will be limited to a diagnostic plot by gpr file and a marrayRaw object.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used. See details for more information.
`controlMatrix`	A character matrix of n by 2 columns. The first column contains a few regular expression of spotted probe sequences and the second column contains the corresponding control status. By default, controlMatrix is set to controlCode.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`output`	Logical. If 'TRUE', normalized M values corresponding to the input GenePix files and quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`val`	A "character" string representing the slotNames to be written in the output file.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Details

gpQuality returns 2 plots for each GenePix files passed as argument. The first one is a qualitative diagnostic plot, a quick visual way to assess slide quality. The second one is a comparative boxplot: each quality control measure is compared to the range obtained for a database of 'good' slides used as reference. You can use your own set of references created using `globalQuality` and `qualRefTable` passed in the arguments "reference" and "scalingTable", or use the reference QC values stored in the datasets `MmReferenceDB` and `MmScalingTable`. All results and quality scores are gathered in a HTML report. For more details about the QC measures and the plots, please refer to the online manual.

Value

A list of 2 elements:

`mraw`	A marrayRaw object created from the input files.
`quality`	A matrix containing Quality Control measures for all slides.

Author(s)

Agnes Paquet

See Also

`globalQuality`, `qualBoxplot`, `scaleRefTable`

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
results <- gpQuality(fnames="9Mm137.gpr", path=datadir, organism="Mm")
```

`gprDB` *Reference slides for Mouse oligos hybridizations*

Description

MmReferenceDB, HsReferenceDB: quality matrix compiling results from a pool of good hybridization slides for Mouse and Human genomes respectively.

MmScalingTable, HsScalingTable: matrix of means and iqr calculated from the pool of good hybridization slides for Mouse and Human genomes respectively, for each quality measure. It is used to scale other arrays quality results.

index.html: HTML file used for quality report.

Source

These data were provided by members of the UCSF Shared functional genomics core lab.

`heeboQuality` *Diagnostics plots designed for HEEBO set controls*

Description

This component generates several exploratory plots customized to the various types of controls provided in the HEEBO set. All results are saved as an image. Tested on GenePix format only (06-29-2006). For more details about the plots, please refer to the HTML description.

Usage

```
heeboQuality(fnames = NULL, path = ".", galfile = NULL, source
="genepix.median", other.columns =
c("Flags"), controlMatrix=HeeboSpotTypes,controlId = c("ID", "Name"),
DOPING = TRUE, heeboSetQC = TRUE, SpotTypeFile = NULL, SpikeTypeFile =
NULL, cy3col = "Cy3_ng", cy5col = "Cy5_ng", id = "SeqID", namecol =
c("Symbol", "Name"), annot = NULL, bgMethod = "none", normMethod = "p",
diagnosticPlot = TRUE, output = TRUE, resdir = ".", dev = "png",
organism = "Hs", DEBUG = FALSE, ...)
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`galfile`	A "character" string naming the file descrining the layout of the array. If missing, meeboQuality will read the layour from the gpr file.
`source`	A "character" string specifing the image analysis program which produced the output files. See ?read.maimages in package limma for more details.
`other.columns`	See ?read.maimages in package limma for more details.
`controlMatrix`	A character matrix of n by 2 or more columns. One column should contain a few regular expression of spotted probe sequences and another column should contain the corresponding control status. By default, controlMatrix is set to HeeboSpotTypes.
`controlId`	Character string. Name of the column of the gpr file (or gal file) used to define controls.
`DOPING`	Logical. If 'TRUE', doping controls quality plots are generated.
`heeboSetQC`	Logical. If 'TRUE', mismatch and tiling controls quality plots are generated.
`SpotTypeFile`	A "character" string representing the name of the file containing spot type description for the array.
`SpikeTypeFile`	A "character" string representing the name of the file containing doping control information. See HTML description for more details.
`cy3col`	A "character" string representing the name of the column of the SpiketypeFile containing the quantity of each control spiked in the Cy3 channel.

`cy5col`	A "character" string representing the name of the column of the SpiketypeFile containing the quantity of each control spiked in the Cy5 channel.
`id`	A "character" string describing which column of the MEEBO annotation should be used to retrieve replicated oligos, e.g. "SeqID".
`namecol`	A "character" string describing which column of the SpiketypeFile should be used in the legend.
`annot`	A "character" string describing which R object should be used to look-up probes annotations. By default, it will be set to HEEBOset.
`bgMethod`	Character string specifying which background correction method to use. See ?backgroundCorrect in package limma for more details.
`normMethod`	Character string specifying which normalization method should be used. See ?normalizeWithinArrays in package limma for more details.
`diagnosticPlot`	Logical. If 'TRUE', a quality diagnostic plot will be generated.
`output`	Logical. If 'TRUE', normalized M values and A values corresponding to the input GenePix files and additionnal quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Hs". It is used to retrieve the corresponding Oligo set annotations.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	Additional arguments

Details

heeboQuality returns 3 types of quality control plots, specifically designed for the various controls offered by the HEEBO set. To assess the global performance of an hybridization, users can generate a diagnostic plot summarizing several graphs and statictics by setting'diagnosticPlot=TRUE'. Then, the performance of the HEEBO set can beanalyzed by looking specifically at the mismatch and the tiling controls ('heeboSetQC=TRUE').

Finally, we are also providing several exploratory tools to assess the performance of the doping-controls ('DOPING=TRUE');these plots should be used only if a spike-in mixture was added to the hybridization solution.

Value

heeboquality will produce several graphs, saved in an image file format. Please refer to the HTML description for more details. The function will also return the MAList object describing your tested slides.

Author(s)

Agnes Paquet

See Also

`heeboQualityPlots`, `gpQuality`, `meeboQuality`

Examples

```
if (interactive())
{
require(HEEBOdata)
datadir <- system.file("Heebo", package="HEEBOdata")
MA <-
heeboQuality(fnames="63421.gpr",galfile="hoc.gal",path=datadir,SpikeTypeFile="DCV2.0June
DOPING=TRUE, heeboSetQC=TRUE, namecol="Name", resdir="HeeboQC", DEBUG=TRUE)
}
```

`heeboQualityPlots` *Qualitative diagnostic plot for general hybridization, specific to*

Description

This function generates diagnostic plots for a qualitative assessment of slide quality.

Usage

```
heeboQualityPlots(mrawObj, headerInfo = "", save = TRUE, dev = "png",
col = NULL, badspotfunction = NULL, controlId = c("ID", "Name"), seqId =
"SeqID", organism = "Hs", DEBUG = FALSE, ...)
```

Arguments

`mrawObj`	`marrayRaw` or `RGList` object representing the slides to be tested.
`headerInfo`	Text to be used as header in the diagnostic plot.
`save`	Logical. If 'TRUE', the plot is saved to a file.
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`col`	Vector of colors to use to describe different categories of spots.
`badspotfunction`	Function to use for bad spots on the array.
`controlId`	Character string. Name of the column of the gpr file used to define controls (or provides the name of the probes).
`seqId`	A "character" string naming the column of the MEEBO annotation to use to retrieve replicated sequences.
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Hs". It is used to retrieve the corresponding Oligo set annotations.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Details

The right panels of the diagnostic plot contain boxplots of the various controls in the set, like positive or negative controls, as set in the marrayRaw of RGlist object. Replicated controls are recognized based on their HEEBO sequence id.

Author(s)

Agnes Paquet

See Also

`heeboQuality`, `gpQuality`, `maQualityPlots`,`HeeboSpotTypes`, `controlCodeHeebo`

Examples

```
if (interactive())
{
require(HEEBOdata)
datadir <- system.file("Heebo", package="HEEBOdata")
gal <- readGAL("hoc.gal", path=datadir)
RG <-
read.maimages(files=c("63421.gpr"), path=datadir, source="genepix.median",other.columns="
RG$genes <- gal
RG$printer <- getLayout(RG$genes)
RG$genes$Status <- controlStatus(HeeboSpotTypes,RG,verbose=TRUE)
rownames(RG$R) <- rownames(RG$G) <- RG$genes[,"ID"]
heeboQualityPlots(RG)
}
```

`maQualityPlots` *Qualitative diagnostic plot for general hybridization*

Description

This function generates diagnostic plots for a qualitative assessment of slide quality.

Usage

```
maQualityPlots(mrawObj, headerInfo = "", save = TRUE, dev = "png", col=NULL,
badspotfunction=NULL, controlId = c("ID", "Name"), DEBUG = FALSE, ...)
```

Arguments

`mrawObj`	`marrayRaw` or `RGList` object representing the slides to be tested.
`headerInfo`	Text to be used as header in the diagnostic plot.
`save`	Logical. If 'TRUE', the plot is saved to a file.
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`col`	Vector of colors to use to describe different categories of spots.
`badspotfunction`	Function to use for bad spots on the array.
`controlId`	Character string. Name of the column of the gpr file used to define controls (or provides the name of the probes).
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Author(s)

Jean Yang

Examples

```
# Example uses swirl dataset...
```

`meeboQuality` *Diagnostics plots designed for MEEBO set controls*

Description

This component generates several exploratory plots customized to the various types of controls provided in the MEEBO set. All results are saved as an image. Tested on GenePix format only (11-18-2005). For more details about the plots, please refer to the HTML description.

Usage

```
meeboQuality(fnames = NULL, path = ".", galfile = NULL, source
="genepix.median", other.columns = c("Flags"),controlMatrix=MeeboSpotTypes,contr
DOPING = TRUE,meeboSetQC = TRUE, SpotTypeFile = NULL, SpikeTypeFile =
NULL, cy3col = "CY3.ng._MjDC_V1.7", cy5col = "CY5.ng._MjDC_V1.7", id =
"SeqID", namecol = c("Symbol", "Name"), annot = NULL, bgMethod = "none", normMet
"p", diagnosticPlot = TRUE, output = TRUE, resdir = ".", dev = "png", organism =
...)
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`galfile`	A "character" string naming the file descrining the layout of the array. If missing, meeboQuality will read the layour from the gpr file.
`source`	A "character" string specifing the image analysis program which produced the output files. See ?read.maimages in package limma for more details.
`other.columns`	See ?read.maimages in package limma for more details.
`controlMatrix`	A character matrix of n by 2 or more columns. One column should contain a few regular expression of spotted probe sequences and another column should contain the corresponding control status. By default, controlMatrix is set to MeeboSpotTypes.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`DOPING`	Logical. If 'TRUE', doping controls quality plots are generated.
`meeboSetQC`	Logical. If 'TRUE', mismatch and tiling controls quality plots are generated.
`SpotTypeFile`	A "character" string representing the name of the file containing spot type description for the array.

`SpikeTypeFile`	A "character" string representing the name of the file containing doping control information. See HTML description for more details.
`cy3col`	A "character" string representing the name of the column of the SpiketypeFile containing the quantity of each control spiked in the Cy3 channel.
`cy5col`	A "character" string representing the name of the column of the SpiketypeFile containing the quantity of each control spiked in the Cy5 channel.
`id`	A "character" string describing which column of the MEEBO annotation should be used to retrieve replicated oligos, e.g. "SeqID".
`namecol`	A "character" string describing which column of the SpiketypeFile should be used in the legend.
`annot`	A "character" string describing which R object should be used to look-up probes annotations. By default, it is set to MEEBOset.
`bgMethod`	Character string specifying which background correction method to use. See ?backgroundCorrect in package limma for more details.
`normMethod`	Character string specifying which normalization method should be used. See ?normalizeWithinArrays in package limma for more details.
`diagnosticPlot`	Logical. If 'TRUE', a quality diagnostic plot will be generated.
`output`	Logical. If 'TRUE', normalized M values and A values corresponding to the input GenePix files and additionnal quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding Oligo set annotations.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	Additional arguments

Details

meeboQuality returns 3 types of quality control plots, specifically designed for the various controls offered by the MEEBO set. To assess the global performance of an hybridization, users can generate a diagnostic plot summarizing several graphs and statictics by setting'diagnosticPlot=TRUE'. Then, the performance of the MEEBO set can beanalyzed by looking specifically at the mismatch and the tiling controls ('meeboSetQC=TRUE').

Finally, we are also providing several exploratory tools to assess the performance of the doping-controls ('DOPING=TRUE');these plots should be used only if a spike-in mixture was added to the hybridization solution.

Value

meeboquality will produce several graphs, saved in an image file format. Please refer to the HTML description for more details. The function will also return the MAList object describing your tested slides.

Author(s)

Agnes Paquet

See Also

`meeboQualityPlots`, `gpQuality`

Examples

```
if (interactive())
{
require(MEEBOdata)
datadir <- system.file("Meebo", package="MEEBOdata")
MA <- meeboQuality(fnames="RDI108_n.gpr",path=datadir,SpikeTypeFile="StanfordDCV1.7comple
DOPING=TRUE, meeboSetQC=TRUE, namecol="Name", resdir="MeeboQC", DEBUG=TRUE)
}
```

`meeboQualityPlots` *Qualitative diagnostic plot for general hybridization, specific to*

Description

This function generates diagnostic plots for a qualitative assessment of slide quality.

Usage

```
meeboQualityPlots(mrawObj, headerInfo = "", save = TRUE, dev = "png",
col = NULL, badspotfunction = NULL, controlId = c("ID", "Name"), seqId =
"SeqID", organism = "Mm", DEBUG = FALSE, ...)
```

Arguments

`mrawObj`	`marrayRaw` or `RGList` object representing the slides to be tested.
`headerInfo`	Text to be used as header in the diagnostic plot.
`save`	Logical. If 'TRUE', the plot is saved to a file.
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`col`	Vector of colors to use to describe different categories of spots.
`badspotfunction`	Function to use for bad spots on the array.
`controlId`	Character string. Name of the column of the gpr file used to define controls (or provides the name of the probes).
`seqId`	A "character" string naming the column of the MEEBO annotation to use to retrieve replicated sequences.
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding Oligo set annotations.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Details

The right panels of the diagnostic plot contain boxplots of the various controls in the set, like positive or negative controls, as set in the marrayRaw of RGlist object. Replicated controls are recognized based on their MEEBO sequence id.

Author(s)

Agnes Paquet

See Also

`meeboQuality`, `gpQuality`, `maQualityPlots`,`MeeboSpotTypes`, `controlCodeMeebo`

Examples

```
if (interactive())
{
require(MEEBOdata)
datadir <- system.file("Meebo", package="MEEBOdata")
mraw <- read.GenePix(path=datadir)
maControls(mraw) <- maGenControls(maGnames(mraw),id="ID",controlcode=controlCodeMeebo)
rownames(maRf(mraw)) <- rownames(maRb(mraw)) <- maGeneTable(mraw)[,"ID"]
rownames(maGf(mraw)) <- rownames(maGb(mraw)) <- maGeneTable(mraw)[,"ID"]
meeboQualityPlots(mraw)
}
```

`prdata` *Example GPR files*

Description

9Mm137.gpr is QCHyb from 9Mm printrun.

12Mm250.gpr is a 9mers hybridization from 12Mm printrun.

Source

These data were provided by members of the UCSF Shared Functional Genomics Core Facility.

`qcScore` *Quality Control score for general hybridization*

Description

This function returns, for each quality measure, the number of qc measures of the tested slides which are below the reference slides boundaries.

Usage

```
qcScore(arrayQuality,reference)
```

Arguments

`arrayQuality`	Matrix of quality results from `slideQuality` or `globalQuality`.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used. See details for more information.

Value

qcScore returns the number of qc measures under the lower limits of the reference values.

Author(s)

Agnes Paquet

`qualBoxplot` *Comparative boxplot for general hybridization Quality Control*

Description

This functions allows you to graphically compare your slide quality measures to results obtained for a database of 'good quality' slides.

Usage

```
qualBoxplot(arrayQuality = NULL, reference = NULL, organism = c("Mm", "Hs"), DEB
```

Arguments

`arrayQuality`	Matrix of quality results from `slideQuality` or `globalQuality`.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to "organism" will be used. See details for more information.
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Details

You can use your own set of references created using `globalQuality` passed in the arguments "reference", or use the reference QC values stored in the datasets `MmReferenceDB`.

Value

Returns a score vector containing, for each column in "arrayQuality", the number of quality measures below the range of "reference". The last element of the score vector is the total number of quality measures tested.

Author(s)

Agnes Paquet

See Also

`globalQuality`, `gpQuality`, `spotQuality`, `agQuality`

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if(interactive())
{
gprData <- readGPR(fnames="9Mm137.gpr", path=datadir)
arrayQuality <- slideQuality(gprData, organism="Mm")
qualBoxplot(arrayQuality)}
```

`qualityScore` *Quality Control score for general hybridization*

Description

This function returns, for each quality measure, the percentage of reference slides measures which are below the tested slide values. For more details on this score, refer to the online manual.

Usage

```
qualityScore(slidequality, organism = c("Mm", "Hs"), reference =
NULL)
```

Arguments

`slidequality`	A quality matrix from slideQuality.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used. See details for more information.
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.

Value

QualityScore returns a matrix of percentages of qc measures under the lower limit of the reference values.

Author(s)

Agnes Paquet

`readAgilent` *Extraction of measures from Agilent (.txt) files.*

Description

This component reads an Agilent file (.txt) and returns columns used for quality control.

Usage

```
readAgilent(fnames = NULL, path= ".", DEBUG=FALSE, skip = 0, sep ="\t",
quote="\"",controlId=c("ProbeName"), ...)
```

Arguments

`fnames`	A "character" string naming the input file.
`path`	a "character" string representing the data directory. By default this is set to the current working directory (".").
`DEBUG`	If 'TRUE', debug statements are printed.
`skip`	Number of lines to skip in the gpr files.
`sep`	A "character" string defining the type of separation for the columns in the gpr files.
`quote`	A "character" string defining the type of quote in the gpr files.
`controlId`	Character string. Name of the column of the file used to define controls.
`...`	additional arguments.

Value

A list of vectors containing information extracted from the Agilent file

Author(s)

Agnes Paquet

See Also

`slideQuality`, `agQuality`, `globalQuality`

`readGPR` *Reading GenePix gpr file*

Description

This component reads a GenePix file (.gpr) and returns columns used for quality control.

Usage

```
readGPR(fnames = NULL, path= ".", DEBUG=FALSE, skip = 0, sep ="\t",
quote="\"", controlId="ID",...)
```

Arguments

`fnames`	A "character" string naming the input file.
`path`	a "character" string representing the data directory. By default this is set to the current working directory (".").
`DEBUG`	If 'TRUE', debug statements are printed.
`skip`	Number of lines to skip in the gpr files.
`sep`	A "character" string defining the type of separation for the columns in the gpr files.
`quote`	A "character" string defining the type of quote in the gpr files.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`...`	additional arguments.

Value

A list of vectors containing information extracted from the GenePix file

Author(s)

Agnes Paquet

See Also

`slideQuality`, `gpQuality`, `globalQuality`

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
gprData <- readGPR(fnames="9Mm137.gpr", path=datadir)
```

`readSpikeTypes` *Read Spike Types File*

Description

Read a table containing information about the doping control mixture used in the hybridization.

Usage

```
readSpikeTypes(file = "DopingTypeFile2.txt", path = NULL, cy5col = "MassCy5", cy
```

Arguments

`file`	A "character" string giving the name of the file specifying the doping controls used.
`path`	A "character" string giving the directory containing the file. By default this is set to the current working directory (".").
`cy5col`	A "character" string describing the name of the column corresponding to the controls labelled with Cy5.
`cy3col`	A "character" string describing the name of the column corresponding to the controls labelled with Cy3.
`...`	Additional arguments passed to "readSpotTypes"

Details

The file is a text file with rows corresponding to doping controls and columns describing various experimental conditions. It must contain an oligo sequence identifier for each control, the spike type (e.g. Ambion or MJ) and the mass of each oligo spiked in each channel. By default, this function assumes that the mass unit are the same.

Value

A list of n matrices, each matrix containing information about a unique type of spiked controls.

Author(s)

Agnes Paquet

Examples

```
datadir <- system.file("Meebo", package="arrayQuality")
if (interactive())
{
spikes <-
readSpikeTypes(file="StanfordDCV1.7complete.txt",path=datadir,cy5col="CY5.ng._MjDC_V1.7",
}
```

`readSpot` *Extraction of measures from Spot (.spot) files.*

Description

This component reads a Spot file (.spot) and returns columns used for quality control.

Usage

```
readSpot(fnames = NULL, path= ".", galfile=NULL,DEBUG=FALSE, skip = 0, sep ="\t"
```

Arguments

`fnames`	A "character" string naming the input file.
`path`	a "character" string representing the data directory. By default this is set to the current working directory (".").
`galfile`	A "character" string naming the galfile associated with the input file. If galfile = NULL, readSpot will use the first .gal file found in the working directory.
`DEBUG`	If 'TRUE', debug statements are printed.
`skip`	Number of lines to skip in the gpr files.
`sep`	A "character" string defining the type of separation for the columns in the gpr files.
`quote`	A "character" string defining the type of quote in the gpr files.
`controlId`	Character string. Name of the column of the Spot file used to define controls.
`...`	additional arguments.

Value

A list of vectors containing information extracted from the Agilent file

Author(s)

Agnes Paquet

See Also

`slideQuality`, `agQuality`, `globalQuality`

`readcontrolCode` *Control status information*

Description

This component reads spot type information from a tab delimited text file to a matrix.

Usage

```
readcontrolCode(file = "SpotTypes.txt", path = NULL, sep = "\t", check.names = F
```

Arguments

`file`	Character string giving the name of the file specifying the spot types.
`path`	Character string giving the directory containing the file. Can be omitted if the file is in the current working directory.
`sep`	the field separator character.
`check.names`	Logical, if 'FALSE' column names will not be converted to valid variable names, for example spaces in column names will not be left as is.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`...`	additional arguments

Value

A 2 column matrix named controlCode.

Author(s)

Jean Yee Hwa Yang, Agnes Paquet

`scaleRefTable` *General hybridization quality scaling*

Description

This function helps you scale quality measures in order to compare them on the same plot. It is used on reference slides to create a look-up table, which will be used to scale other slides.

Usage

```
scaleRefTable(reference=NULL, organism=c("Mm", "Hs"))
```

Arguments

`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used.

Value

A matrix containing median and iqr for each quality measure for tested slides.

Author(s)

Agnes Paquet

See Also

`gpQuality`, `globalQuality`, `qualBoxplot`

`slideQuality` *Quality Control statistics for general hybridization*

Description

This component uses data extracted from GenePix file to provide quality control statistics.

Usage

```
slideQuality(gprData = NULL, controlMatrix = controlCode, controlId = c("ID", "N
```

Arguments

`gprData`	A list of vector, results from `readGPR`, containing information extracted from the gpr file.
`controlMatrix`	A matrix defining control status information.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Value

A matrix of numbers.

Author(s)

Agnes Paquet

See Also

`gpQuality`, `globalQuality`, `readGPR`

Examples

```
datadir <- system.file("gprQCData", package="arrayQuality")
if (interactive())
{
gprdata <- readGPR(fnames="9Mm137.gpr", path=datadir)
results <- slideQuality(gprdata)
}
```

`spotQuality` *Diagnostic plots and comparative boxplots for general hybridization,*

Description

This component provides qualitative diagnostic plots and quantitative measures for assessing general hybridization quality. All results are displayed in a HTML report. Spot format only.

Usage

```
spotQuality(fnames = NULL, path = ".", galfile = NULL, organism = c("Mm", "Hs"),
compBoxplot = TRUE, reference = NULL, controlMatrix = controlCode,
controlId = c("ID"), output = FALSE, resdir = ".", dev= "png", DEBUG = FALSE,...
```

Arguments

`fnames`	A "character" string naming the input files.
`path`	A "character" string representing the data directory. By default this is set to the current working directory (".").
`galfile`	A "character" string naming the galfile associated with the input files. If galfile = NULL, spotQuality will use the first .gal file available in the working directory.
`organism`	A "character" string naming the organism genome printed on the array, either "Mm" or "Hs". By default, organism is set to "Mm". It is used to retrieve the corresponding reference tables.
`compBoxplot`	Logical. If set to 'FALSE', only qualitative diagnostic plots will be plotted. gpQuality ouput will be limited to a diagnostic plot by gpr file and a marrayRaw object.
`reference`	A matrix resulting from globalQuality, to be used as reference table to compare slides. If 'NULL', the default table corresponding to organism will be used. See details for more information.
`controlMatrix`	A character matrix of n by 2 columns. The first column contains a few regular expression of spotted probe sequences and the second column contains the corresponding control status.
`controlId`	Character string. Name of the column of the gpr file used to define controls.
`output`	Logical. If 'TRUE', normalized M values corresponding to the input GenePix files and quality measures are printed to a file.
`resdir`	A "character" string representing the directory where the results will be saved. By default, this is set to the current working directory (".").
`dev`	A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".
`DEBUG`	If 'TRUE', debug statements are printed.
`...`	additional arguments

Details

agQuality returns 2 plots for each Agilent files passed as argument. The first one is a qualitative diagnostic plot, a quick visual way to assess slide quality. The second one is a comparative boxplot: each quality control measure is compared to the range obtained for a database of 'good' slides used as reference. You can use your own set of references created using `globalQuality` passed in the arguments "reference", or use the reference QC values stored in the datasets `MmReferenceDB` and `HsReferenceDB`. All results and quality scores are gathered in a HTML report. For more details about the QC measures and the plots, please refer to the online manual.

Value

A list of 2 elements:

`mraw`	A marrayRaw object created from the input files.
`quality`	A matrix containing Quality Control measures for all slides.

Author(s)

Agnes Paquet

See Also

`globalQuality`, `qualBoxplot`, `readAgilent`

Index