RNAinteract

October 25, 2011

`RNAinteract-class` *Class "RNAinteract"*

Description

A class for double perturbation experiments (genetic interaction screens, drug-drug interaction screens). There are functions for creation, analysis, and display of interaction screens.

Objects from the Class

Objects can be created by calls of `createRNAinteractFromFiles`. See vignette("RNAinteract") for an example of creating an `RNAinteract` object.

Slots

`data`: Object of class `"array"` with dimension `sgi@F x sgi@S x sgi@C`. The raw data of the screen.

`screenNames`: Object of class `"character"` with length `sgi@S`.

`channelNames`: Object of class `"character"` with length `sgi@C`.

`well`: Object of class `"character"` with length `sgi@F`. Well name (e.g. F04) for each measurement.

`plate`: Object of class `"integer"` with length `sgi@F`. Number of the plate for each measurement

`pdim`: Object of class `"integer"` of length 2. Plate dimensions.

`NT`: Object of class `"integer"` of length 1. Number of template reagents.

`NQ`: Object of class `"integer"` of length 1. Number of query reagents.

`C`: Object of class `"integer"` of length 1. Number of readout channels.

`S`: Object of class `"integer"` of length 1. Number of screens.

`F`: Object of class `"integer"` of length 1. Number of measurements or single experiments per screen.

`reagents`: Object of class `"data.frame"` describing each reagents. Obligatory columns: `RID` and `TID`.

`targets`: Object of class `"data.frame"` describing each target gene. Obligatory columns: `TID`, `Symbol`, `group`, `GID`.

`templateDesign`: Object of class `"data.frame"` with `sgi@NT` rows describing the template design. Obligatory columns: `TemplatePlate`, `Well`, `RID`, `QueryNr`.

`queryDesign`: Object of class `"data.frame"` with `sgi@NQ` rows describing the query design. Obligatory columns: `Plate`, `TemplatePlate`, `QueryNr`, `RID`.

`transformation`: Object of class `"character"` of length `sgi@C`. The transformation applied to the input data.

`mainTemplate`: Object of class `"array"` with dimension `sgi@NT x sgi@S x sgi@C`. The main effect of the template reagents.

`mainQuery`: Object of class `"array"` with dimension `sgi@NQ x sgi@S x sgi@C`. The main effect of the query reagents.

`mainSderrTemplate`: Object of class `"array"` with dimension `sgi@NT x sgi@S x sgi@C`. The standard error of the main effect of the template reagents.

`mainSderrQuery`: Object of class `"array"` with dimension `sgi@NQ x sgi@S x sgi@C`. The standard error of the main effect of the query reagents.

`mainSdTemplate`: Object of class `"array"` with dimension `sgi@NQ x sgi@S x sgi@C`. The standard deviation of the main effect of the query reagents.

`mainSdQuery`: Object of class `"array"` with dimension `sgi@NQ x sgi@S x sgi@C`. The standard deviation of the main effect of the query reagents.

`mainTimeEffect`: Object of class `"array"` with dimension `sgi@NQ x sgi@S x sgi@C`. The systematic changes of the query main effects, e.g. decreasing cell number over time.

`mainSpatialEffect`: Object of class `"array"` with dimension `sgi@F x sgi@S x sgi@C`. The systematic spatial plate effects.

`mainSpatialEffectRow`: Object of class `"array"`. Spatial effects per row (as computed by Bscore).

`mainSpatialEffectCol`: Object of class `"array"`. Spatial effects per column (as computed by Bscore).

`mainNeg`: Object of class `"array"` with dimension `sgi@S x sgi@C`. The main effect of the negative control.

`mainNegTemplate`: Object of class `"array"` with dimension `sgi@S x sgi@C`. The template main effect of the negative control.

`mainNegQuery`: Object of class `"array"` with dimension `sgi@S x sgi@C`. The query main effect of the negative control.

`data2mainTemplate`: Object of class `"integer"` with dimension `sgi@F`. Mapping of single experiments to template reagents.

`data2mainQuery`: Object of class `"integer"` with dimension `sgi@F`. Mapping of single experiments to query reagents.

`ni.model`: Object of class `"array"` with dimension `sgi@F x sgi@S x sgi@C`. The expected values of the non-interacting model.

`pi`: Object of class `"array"` with dimension `sgi@F x sgi@S x sgi@C`. The pairwise interaction score.

`plateeffect`: Object of class `"array"`.

`p.value`: Object of class `"array"` with dimension `sgi@NT x sgi@NQ x sgi@S x sgi@C` describing the p.values.

`q.value`: Object of class `"array"` with dimension `sgi@NT x sgi@NQ x sgi@S x sgi@C` describing the q.values.

Methods

show `signature(object = "RNAinteract")`: ...

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
showClass("RNAinteract")
```

`RNAinteract-package` *Analysis of Pairwise Interaction Screens.*

Description

The package contains functions to organize the data from (single- and multi-parametric) genetic interaction screens. Methods to estimate main effects (single perturbation effects) and pairwise interactions. p-values are computed. Furthermore a comprehensive html-report is generated.

Details

See vignette("RNAinteract") for details.

Package content

Class `RNAinteract` (Documentation: `RNAinteract-class`)

Data input and creating of an object of class `RNAinteract`.

- `createCellHTSFromFiles`
- `createRNAinteract`, `createRNAinteractFromFiles`

Data access

- `getData` Primary data access function for multiple types of screen data.
- `getMain`, `getMainNeg` access to main effects.
- `getReplicateData`, `getIndDesignData` Comparing replicate data.
- `getChannelNames`, `getScreenNames`, `getScale`

Subsetting, summarizing, and binding screens

- `sgisubset`, `sgisubsetQueryDesign`
- `bindscreens`
- `summarizeScreens`

Main effects and pairwise interactions

- estimateMainEffect
- normalizeMainEffectQuery, normalizeMainEffectTemplate, normalizePlateEffect
- computePI, computePValues
- embedPCA

Plotting

- plotDoublePerturbation, plotHeatmap standard plot functions
- doublePerturbationGrob, grid.doublePerturbation, grid.sgiHeatmap specialized grid plotting functions for experts

HTML report

- startReport, endReport starting and finalizing a report
- reportAnnotation, reportStatistics global reports
- reportDoublePerturbation, reportGeneLists, reportHeatmap, reportMainEffects, reportNetworks, reportScreenData reports specific for each screen and each channel

Author(s)

Bernd Fischer

Maintainer: Bernd Fischer <bernd.fischer@embl.de>

References

T. Horn, T. Sandmann, B. Fischer, W. Huber, M. Boutros. Mapping of Signalling Networks through Synthetic Genetic Interaction Analysis by RNAi. Nature Methods, 2011.

bindscreens *bind RNAinteract objects along screens*

Description

Bind two RNAinteract objects along screens.

Usage

```
bindscreens(sgi1, sgi2)
```

Arguments

sgi1	An object of class RNAinteract.
sgi2	An object of class RNAinteract.

Details

This function binds two double interaction screens along screens.

Value

An object of class RNAinteract with all screens in sgi1 and sgi2.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

RNAinteract-package

Examples

```
data("sgi")
sgi
sginew <- summarizeScreens(sgi, screens=c("1","2"), newscreenname = "m")
sginew
sgibind <- bindscreens(sgi, sginew)
sgibind
```

---

computePI *compute pairwise interaction*

Description

Compute the pairwise interaction term for each single experiments.

Usage

```
computePI(sgi)
```

Arguments

sgi	An object of class RNAinteract.

Details

Computes the pairwise interaction term for each single experiment. Multiple values for each gene pair are not yet summarized.

Value

An object of class RNAinteract.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

RNAinteract-package

Examples

```
data("sgi")
sgi <- computePI(sgi)
PI <- getData(sgi, type="pi", format="targetMatrix")
```

computePValues *compute p-values*

Description

Compute p-values for genetic interactions terms. Assess if genetic interaction term is different from zero.

Usage

```
computePValues(sgi, method = "pooled.ttest", mixTemplateQuery = TRUE, verbose =
```

Arguments

`sgi` An object of class RNAinteract.

`method` The method used to compute p-values. One of "pooled.ttest","ttest", "limma", "HotellingT2".

For "ttest" a Student t-test is applied for each gene pair. The variance is estimated locally for each gene pair. For "pooled.ttest", a pooled variance is estimated from all gene pairs. The variance applied for each gene pair is the maximum of the pooled and the local variance estimate. This method obtains conservative p-values. For "limma" mediates between the local and the global variance estimation in a Bayesian framework. The limma-package is applied to compute the p-values. For "HotellingT2" Hotelling-T^2 statistics is computed jointly for all dimensions. It results in a single p-value summarizing all channels. For simplification the p-values are stored in a matrix of dimension genes x genes x screens x channels and the p-values are repeated for each channel. The same holds for q-values.

`mixTemplateQuery`

If a gene-pair is measured twice as template-query and as query-template, a single p-value is computed by combining all measurements, if `mixTemplateQuery = TRUE`. Else a p-value is computed independently for both cases.

`verbose` Either 0 (default, no output), 1 (minimum output), or 2 (outout)

Details

Computes p-values from a t-test, using the bioconductor package limma, or with a multidimensional Hotelling T^2 test.

Value

An object of class `RNAinteract`.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

`RNAinteract-package`

Examples

```
data("sgi")
sgi <- computePValues(sgi, method = "HotellingT2")
# Hotelling T^2 test will provide one p-value for all channels, PV will be the same
# for all channels in this case
PV <- getData(sgi, type="p.value", format="targetMatrix", channel="nrCells")
```

`createCellHTSFromFiles` *create cellHTS2 object from text files*

Description

A cellHTS2 object is created from a set of text files.

Usage

```
createCellHTSFromFiles(filePlatelist = "Platelist.txt",
                              name = "anonymous",
        path = ".", pdim = NULL)
```

Arguments

`filePlatelist`
The platelist (See vignette("RNAinteract"))

`name` A (arbtrary) string providing the name for the screen

`path` The path were the data files are located

`pdim` Giving the plate dimensions, e.g. pdim=c(8,12) or pdim=c(16,24).

Details

See vignette("RNAinteract") for an example how to create an RNAinteract object.

Value

An object of class `cellHTS2`.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

RNAinteract-package, createRNAinteractFromFiles, createRNAinteract

createRNAinteract *create a RNAinteract object*

Description

Creates a RNAinteract object given data matrices, annotation, query and template design.

Usage

```
createRNAinteract(data, well, plate, pdim,
                  Reagents, Targets,
                        TemplateDesign, QueryDesign,
                        Transformation = NULL)
```

Arguments

`data` An array with dimensions features x screens x channels.

`well` A vector of length #features with well names.

`plate` A vector of length #features with plate numbers.

`pdim` A vector of length 2 with plate dimensions (e.g. pdim=c(12,8)).

`Reagents` A data.frame describing the reagents.

`Targets` A data.frame describing the targets.

`TemplateDesign` A data.frame with the layout of the template plates.

`QueryDesign` A data.frame with the layout of the query plates.

`Transformation` A Transformation that is applied to the data. If NULL the data is log2 transformed.

Details

See vignette("RNAinteract") for an example how to create an RNAinteract object.

Value

An object of class RNAinteract.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

`RNAinteract-package`, `createRNAinteractFromFiles`, `createCellHTSFromFiles`

`createRNAinteractFromFiles` *create an RNAinteract object from text files*

Description

Reads text files with annotation, query and template design, and data. Creates a RNAinteract object.

Usage

```
createRNAinteractFromFiles(name = "anonymous",
  filePlatelist = "Platelist.txt",
      fileReagents = "Reagents.txt",
      fileTargets = "Targets.txt",
      fileTemplateDesign = "TemplateDesign.txt",
      fileQueryDesign = "QueryDesign.txt",
      path = ".",
      pdim = NULL,
      Transformation = "log2")
```

Arguments

`name` A name for the screen.

`filePlatelist` The filename of the text file containing the plate list.

`fileReagents` The filename of the text file containing the reagent annotation.

`fileTargets` The filename of the text file containing the target annotation.

`fileTemplateDesign` The filename of the text file containing the template design.

`fileQueryDesign` The filename of the text file containing the query design.

`path` The system directory were the textfiles `filePlatelist`, `fileReagents`, `fileTargets`, `fileTemplateDesign`, and `fileQueryDesign` are located.

`pdim` The dimensions of the multi-well plates (e.g. `pdim = c(nrow=24,ncol=16)`). if `pdim =NULL` (default), the plate dimension will be estimated from the input data.

`Transformation` The transformation that is applied to the data. All calculations are done on additive scale.

Details

See vignette("RNAinteract") for an example how to create an RNAinteract object.

Value

An object of class `RNAinteract`.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

`RNAinteract-package`, `createRNAinteract`, `createCellHTSFromFiles`

`embedPCA` *Computes a PCA for a pairwise interaction matrix.*

Description

A principal component analysis is performed for a pairwise interaction matrix. The low-dimensional embedding is returned.

Usage

```
embedPCA(sgi, screen, channel,
         dim = 4, embed = "template",
         withoutgroups = c())
```

Arguments

`sgi`	An object of class `RNAinteract`
`screen`	The screen name whose interaction matrix will be embedded.
`channel`	The channel name whose interaction matrix will be embedded.
`dim`	The embedding dimension.
`embed`	Either "template" (default) or "query" denotes if the embedding is done for rows or columns.
`withoutgroups`	Genes annotated with these groupnames are not considered for embedding.

Value

Returns a matrix with dimensions genes x dim.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
X <- embedPCA(sgi, screen="1", channel="nrCells", dim=2)
plot(X[,1], X[,2], pch=20, cex=0.01)
text(X[,1], X[,2], row.names(X))
```

---

estimateMainEffect *estimate main effect*

Description

estimates the main effects in an additive model.

Usage

```
estimateMainEffect(sgi, use.query = NULL)
```

Arguments

sgi	An object of class RNAinteract.
use.query	A list of reagent identifiers as annotated in the RID field of the reagent list. For the estimation of the template main effects only these queries are used.

Details

The main effect is the single RNAi knockdown effect. When use.query is not specified, the main effect is estimated by minimizing the L1 distance from the non-interacting model to the double RNAi measurements. The implemented non-interacting model is the additive model (sum of single main effects). If the screen does not contain a lot of query genes with no or ery small main effect, it is recommended to estimate the template main effects only by using selected query genes. This can be obtained by specifying use.query. To estimate main effects in a multiplicative model define `Transformation="log2"` when creating the RNAinteract object (See `createRNAinteractFromFiles`), which is already the default.

Value

An object of class RNAinteract.

Author(s)

Bernd Fischer

References

~put references to the literature/web site here ~

See Also

RNAinteract-package

Examples

```
data("sgi")
sgi <- estimateMainEffect(sgi)
getMain(sgi)
```

---

getData *Primary access function for all screen data.*

Description

This function is the primary access function for a wide range of data from the screen. It does perform normalization, transformation, and reshaping if specified.

Usage

```
getData(sgi, type = "data", format = "plain",
design = "template", mixTemplateQuery = TRUE,
screen = NULL, channel = NULL,
do.trafo = TRUE, do.inv.trafo = FALSE,
normalized = FALSE, withoutgroups = c(),
drop = TRUE)
```

Arguments

`sgi` An object of class RNAinteract

`type` Specifies which data is returned. Possible values are:

- "p.value", "q.value": returns the p-value or q-value as computed by computePValues.
- "data": returns the input data.
- "pi": returns the pairwise interaction score.
- "plateeffect": returns the plate effect estimated by normalizePlateEffect.
- "ni.model" returns the non-interacting model as estimated by estimateMainEffect.
- "main": returns the main effects.
- "mainsderr": returns the std error of the main effects.
- "mainsd": returns the std deviation of the main effects.
- "maintime": returns the estimated time effect as estimated by normalizeMainEffectQuery
- "mainspatial": returns the estimated spatial effect as estimated by normalizePlateEffect

`format` The output format. Possible values:

- "plain": The data can be returned as a plain vector
- "platelist": a list of plate matrices that can be passed to plotScreen
- "reagentMatrix": All values for the same reagent pair are summarized in a matrix of dimension reagents x reagents
- "targetMatrix": All values for the same gene pair are summarized in a matrix of dimension genes x genes

`design` If type is one of the main effect types, the design can be specified to state if the "template" or "query" main effect is returned.

`mixTemplateQuery` If TRUE, The template-query and query-template entries in the matrix are symmetrized.

`screen` The screen names of which data should be returned.

`channel` The channel names of which data should be returned.

`do.trafo` Only effective, if type is "data". If TRUE, the data is transformed.

`do.inv.trafo` Not effective if type is "data", "p.value", or "q.value". If TRUE, the values are back-transformed to the original scale.

`normalized` If TRUE, the normalization data is returned.

`withoutgroups` The genes from the specified groups are not returned in the data.

`drop` If FALSE, the returned array is reduced in dimensions, whenever there is a dimension 1.

Value

An array containing the specified values is returned. In the case, the format is chosen to be "platelist", a list of matrices is returned.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")

# get the original data, as plain file, reshaped in plate layout,
# reshaped and summarized as target matrix
D <- getData(sgi, type="data", do.inv.trafo = TRUE)
Dplatelayout <- getData(sgi, type="data",
     format="platelist", do.inv.trafo = TRUE)
splots::plotScreen(Dplatelayout[["1"]][["nrCells"]],
     nx=sgi@pdim[2], ny=sgi@pdim[1], ncol=3)
Dmatrix <- getData(sgi, type="data",
     format="targetMatrix", do.inv.trafo = TRUE)

# get main effects as plate layout with specified transformation
# (usually log-transformed)
Mplatelayout <- getData(sgi, type="main", design="template",
     screen="1", channel="nrCells", format="platelist")
splots::plotScreen(Mplatelayout, nx=sgi@pdim[2], ny=sgi@pdim[1],
     ncol=3)

# get non-interacting model and pairwise interaction scores as matrix
NImatrix <- getData(sgi, type="ni.model", format="targetMatrix")
PImatrix <- getData(sgi, type="pi", format="targetMatrix")
```

```
PIplatelayout <- getData(sgi, type="main", design="query",
     screen="1", channel="nrCells", format="platelist")
splots::plotScreen(PIplatelayout, nx=sgi@pdim[2], ny=sgi@pdim[1],
             ncol=3)

# get p-values and q-values
PVmatrix <- getData(sgi, type="p.value", format="targetMatrix")
QVmatrix <- getData(sgi, type="q.value", format="targetMatrix")
```

getMain *get main effects*

Description

Returns the main effects.

Usage

```
getMain(sgi, type = "main", design = "template", summary = "none",
         QueryNr = NULL, TemplatePlate = NULL,
do.inv.trafo = FALSE, format = "plain", withoutgroups = c(),
screen = NULL, channel = NULL, normalized = TRUE, drop = TRUE)
getMainNeg(sgi, type = "all", do.inv.trafo = FALSE,
screen = NULL, channel = NULL, drop = TRUE)
```

Arguments

`sgi` An object of class RNAinteract

`type` always "main"

`design` Either "template" or "query" defining if template or query main effects are returned.

`summary` If summary is "targets" the main effects are summarized per target gene.

`QueryNr,TemplatePlate` Onle main effects of one query nr or one template plate are returned.

`format` targetmatrix

`withoutgroups` The genes within this group are not shown in the heatmap. It is convenient to hide screen controls.

`do.inv.trafo` If TRUE, the data will be back-transformed for original scale of data. In the case of log-transformed data, the main effects are returned as factors, otherwise the main effects are returned as log values.

`screen` The screen from which the main effects should be returned.

`channel` The channel from which the main effects should be returned.

`drop` Does return a drop array dimensions, even if only one screen or one channel is selected.

`normalized` If true the normalized main effects are returned.

Value

An array containing the main effects.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")
getMain(sgi)
getMainNeg(sgi)
```

`getReplicateData` *Extract replicates measurements from the screen.*

Description

A genetic interaction screen can contain within screen replicates, if some reagent pairs are measured at least twice. Usually this appears when measuring reagent pairs once as template-query and once as query-template. `getReplicateData` returns a list of these technical replicates.

If multiple reagents are used to target the same gene, different reagent pairs that target the same gene pair are extracted from the screen. These pairs are returned by `getIndDesignData`.

Usage

```
getReplicateData(sgi, screen, channel,
                 type = "data", design = "template",
                 do.trafo = TRUE, do.inv.trafo = FALSE,
                 normalized = FALSE)
getIndDesignData(sgi, screen, channel,
                 type = "data", design = "template",
 do.trafo = TRUE, do.inv.trafo = FALSE,
 normalized = FALSE)
```

Arguments

`sgi` An object of class `RNAinteract`.

`screen` The screen name from which the replicates will be extracted.

`channel` The channel name from which the replicates will be extracted.

`type` The type of data that is extracted. It is the type argument of the `getData` function.

`design, do.trafo, do.inv.trafo, normalized`
See the `getData` documentation for details.

Value

Returns a data.frame with columns x and y.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
res <- getIndDesignData(sgi, screen="1", channel="nrCells", type = "data")
plot(res$x, res$y)
```

getScreenNames *get names of screens and channels*

Description

Returns the names of all screens or all channels.

Usage

```
getScreenNames(sgi)
getChannelNames(sgi)
```

Arguments

sgi	RNAinteract

Value

A vector of screen or channel names.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
getScreenNames(sgi)
getChannelNames(sgi)
```

getScale *get scale information for a channel.*

Description

Returns a character string with the scale of each channel.

Usage

```
getScale(sgi, channel)
```

Arguments

sgi	A RNAinteractobject.
channel	A channel name.

Value

Returns a character string with scale information for each channel.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
getScale(sgi, channel="nrCells")
```

grid.doublePerturbation *Double Perturbation Plot Grob*

Description

These functions create a double perturbation grob for interaction screens. All interactions of one gene are displayed in one panel. The double perturbation readout level is plotted against the single perturbation level.

Usage

```
doublePerturbationGrob( mainEffect, dpEffect, mainEffectTarget,
                       range=NULL, main=NULL, xlab=NULL, ylab=NULL,
                       text=NULL, avoid.overlap=TRUE,
                       axisOnOrigin = FALSE,
                       drawBox = TRUE,
       pch = 21, size=unit(1, "char"), fill = NULL,
                       gpMain = gpar(lty="dashed", lwd=3, col="cyan"),
                       gpNI = gpar(lty="dashed", lwd=3, col="orange"),
                       gpPoints = gpar(pch=21),
                       gpText = NULL,
                       gpAxis = NULL,
                       gpWTLines=NULL,
                       name=NULL, gp=NULL, vp=NULL )
grid.doublePerturbation(..., draw = TRUE)

# a helper function for doublePerturbationGrob:
postDrawDetails.doublePerturbation(x, recording)
```

Arguments

`mainEffect` A numeric vector of main effects.

`dpEffect` A numeric vector of double perturbation effects.

`mainEffectTarget` The main effect of the target gene (A single numeric value).

`range` The range of the plot. Equals the `xlim`, `ylim` arguments of `plot`.

`main` An overall title of the plot.

`xlab` A title of the x-axis.

`ylab` A title of the y-axis.

`text` A character vector of text. Has to have the same length as `mainEffect`.

`avoid.overlap` If `TRUE` (default) the text labels are moved such that the text is not overlapping.

`axisOnOrigin` If `TRUE`, the x- and y-axis are draw on the origin of the data. If `FALSE` (default), the axis are drawn on the left and on the bottom.

`drawBox` If `TRUE` (default), a box is drawn around the plot.

`pch` Either an integer specifying a symbol or a single character to be used in plotting points. See `points` for possible values.

`size` A unit object specifying the size of the plotting symbols.

`fill` A list containing (some of) the following elements: `col` defines a fill color for the points. Either a single value or a vector of the same length as `mainEffect`. If `col` is defined, all other elements of fill have no effect. `values` is a numeric vector of the same length as `mainEffcet` that contains values that are mapped to colors. `at` is a numeric vector indicating breakpoints along the values. If not specified will be equally spaced on the range of the values. `colors` defines a set of colors from which a colorramp is created. `colramp` defines a colorramp directly. `colramp` has no effect, if `colors` is defined.

`gp` An object of class `gpar`, typically the output from a call to the function `gpar`. This is basically a list of graphical parameter settings. Overall settings for the plot are set in `gp`.

`gpMain,gpNI` An object of class `gpar` (See `gp`). `gpMain` and `gpNI` indicate the graphics parameter for the main effect lines and the non-interacting line.

`gpPoints,gpText,gpAxis,gpWTLines`
An object of class `gpar` (See `gp`). These arguments define graphical parameters for single compartments of the plot.

`name` A character identifier.

`vp` A Grid viewport object (or NULL).

`draw` If `TRUE` the grob is drawn on the current device.

`...` Further arguments passed to `doublePerturbationGrob`.

`x, recording` Internal usage only.

Details

This function creates a grob for a double perturbation plot. It is probably more convenient to use the function `plotDoublePerturbation`.

Value

A grob is returned.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`, `plotDoublePerturbation`, `reportDoublePerturbation`

`grid.sgiHeatmap` *A heatmap grob*

Description

A grob is created and printed for a matrix PI which is intended to represent pairwise interaction scores.

Usage

```
grid.sgiHeatmap(PI, pi.max = NULL, main = expression(paste(pi, "-score")),
hc.row = NULL, hc.col = NULL)
```

Arguments

`PI` A matrix of pairwise interactions.

`pi.max` The interaction score at the top end of the colorbar. pairwise interaction score larger than this value can not be distinguished anymore.

`main` A title for the plot.

`hc.row` An hierarchical clustering as produced by hclust of the rows.

`hc.col` Clustering of the columns.

Details

A heatmap is plotted with positive interaction represented in yellow and negative interactions represented in blue. A colorbar is plotted on the left and dendrograms are added. This function can be used to integrate the plot in other grid objects. It is recommended to use the function `plotHeatmap` to plot heatmaps of an `RNAinteract` object.

Value

A grob is returned.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")
PI = getData(sgi, type="pi", format="targetMatrix", screen="1", channel="nrCells")
grid.sgiHeatmap(PI)
```

---

normalizeMainEffectQuery

normalize query main effect

Description

Normalize for a time effect of the query genes.

Usage

```
normalizeMainEffectQuery(sgi, batch = NULL, time = NULL)
```

Arguments

`sgi`	An object of class `RNAinteract`.
`batch`	batch is a vector if integers with length equal to the number of queries. It assigns each query to a batch. Within each batch a linear regression is estimated assuming a linear effect between the order of queries and the main effects.
`time`	batch is a vector of numbers. A linear regression is estimated fitting the main effect as a function of the time.

Details

Normalizing the query main effect does not influence the estimation of the pairwise interaction term.

Value

An object of class RNAinteract.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
sgi <- normalizeMainEffectQuery(sgi)
```

normalizeMainEffectTemplate
normalize template main effect

Description

Normalize for a spatial main effect of the template genes.

Usage

```
normalizeMainEffectTemplate(sgi, screen = NULL, channel = NULL)
```

Arguments

sgi	An object of class RNAinteract.
screen	The name of the screen in which the normalization should be applied. If screen = NULL, the normalization is applied on all screens.
channel	The name of the channel in which the normalization should be applied. If channel = NULL, the normalization is applied on all channels.

Details

Normalizing the query main effect does not influence the estimation of the pairwise interaction term.

Value

An object of class RNAinteract.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
sgi <- normalizeMainEffectTemplate(sgi)
```

`normalizePlateEffect` *Normalization of plate effects*

Description

Normalization of plate effects in the screen.

Usage

```
normalizePlateEffect(sgi, type = "Bscore", maxit = 20, verbose = 0)
```

Arguments

`sgi`	An object of class `RNAinteract`
`type`	If type is "Bscore" (default) a Bscore-normalization is performed. If type is "spatial", a locfit regression is estimated that accounts for spatial effects.
`maxit`	Maximum number of iterations for locfit.
`verbose`	Either 0 (default, no output), 1 (minimum output), or 2 (outout).

Details

The Bscore normalization estimates row and column effects for each plate. It returns the residuals to the sum of row and column effects. The spatial normalization estimates a non-linear 2D regression for each plate and returns the residuals.

Value

An object of class `RNAinteract`. The returned object contains the normalization information.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")
normalizePlateEffect(sgi)
```

plotDoublePerturbation

Double Perturbation Plot

Description

These function draws a double perturbation plot for interaction screens. All interactions of one gene are displayed in one panel. The double perturbation readout level is plotted against the single perturbation level.

Usage

```
plotDoublePerturbation( sgi, screen, channel, target,
                       withoutgroups = c("neg", "pos"), design,
                       main, xlab, ylab, range,
                       show.labels = "none", label.par, label,
       avoid.overlap, col, fill,
                       D , MT, MQ, PV, QV, PI, ...)
```

Arguments

`sgi` An object of class `RNAinteract`.

`target` A character name of the target gene.

`screen` The character name of the screen to display. If not specified, the first screen is used. Does not have to be specified, if `sgi` contains only one screen.

`channel` The character name of the channel to display. If not specified, the first channel is used. Does not have to be specified, if `sgi` contains only one channel.

`withoutgroups` Interactions to genes from these groups (as specified in the reagent or target annotation) are excluded from the plot, e.g. positive and negative controls.

`design` The Either `"template"` (default) or `"query"`. The single perturbation effects are either the template main effects or the query main effects.

`main` An overall title of the plot.

`xlab` A title of the x-axis.

`ylab` A title of the y-axis.

`range` A numeric vector of length two. range equals the `xlim`, `ylim` argument in plot.

`show.labels` Automatically select text labels for the points. `'all'` shows a text label for all genes, `"q.value"` and `"p.value"` show a text label for all genes with a q.value (p.value) larger than `label.par`, `"none"` does not show any text label. This argument has no effect, if `label` is specified.

`label.par` Cut-off value for q.value or p.value for displaying text labels (See `show.labels`).

`label` Either a character vector with gene names, or a named vector of text labels. The names of the vector represent the gene names.

`avoid.overlap` If `TRUE` (default), text is moved such that text labels are not overlapping.

`col` A named vector with colors. The names of `col` define which points are colored (See also `fill`).

`fill` A list up to four values. `colors` defines a set of colors from which a colorramp is created. If `colramp` is specified, colors has no effect. `colramp` directly specifies the colorramp. `values` define the values that are color coded. If `values` is not specified, the pairwise interaction term is used instead. `at` is a numeric vector defining the breakpoints along the values. If not specified, breakpoints are selected to range three times the standard deviation of the values around zero. `fill` has no effect, if `col` is specified.

`D,MT,MQ,PV,QV,PI`
Internal usage.

`...` Further argument passed to `grid.doublePerturbation` or `doublePerturbationGrob`.

Details

Plots a double perturbation plot. It shows the interaction profile for one (query) gene.

Value

A grob is returned.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`, `grid.doublePerturbation`, `reportDoublePerturbation`

Examples

```
data("sgi")
plotDoublePerturbation( sgi, screen="1", channel="nrCells", target="rl", show.labels="p.v
```

`plotHeatmap` *plots a heatmap for an interaction screen.*

Description

A heatmap of an interaction screen is plotted.

Usage

```
plotHeatmap(sgi, screen, channel, pi.max = NULL,
            main = expression(paste(pi, "-score")),
    hc.row = NULL, hc.col = NULL,
    withoutgroups = c("neg", "pos"))
```

Arguments

sgi	An object of class RNAinteract
screen	The screen name of which the interaction matrix is plotted.
channel	The channel name of which the interaction matrix is plotted.
pi.max	The pairwise interaction score that is represented at the top of the color scale. All interaction scores above this value can not be distinguished any more.
main	The title of the plot.
hc.row	A hierarchical clustering (hclust) for the rows.
hc.col	A hierarchical clustering (hclust) for the columns.
withoutgroups	The genes within this group are not shown in the heatmap. It is convinient to hide screen controls in the heatmap.

Details

A heatmap for one screen and one channel is plotted. Positive interactions are marked blue, negative ones are marked yellow. A colorbar is shown on the left hand side.

Value

Returns a grob.

Author(s)

Bernd Fischer

See Also

RNAinteract-package

Examples

```
data("sgi")
plotHeatmap(sgi, screen="1", channel="nrCells")
```

---

reportAnnotation *Specialized report functions*

Description

Functions that provide a html report of genetic interactions screens for specific topics.

Usage

```
reportAnnotation        (sgi, verbose = 0, path = ".", dir = "annotation",
                         prefix = "annotation", report = NULL)
reportStatistics        (sgi, verbose = 0, path = ".", dir = "stats",
 prefix = "stat", report = NULL)
reportGeneLists         (sgi, verbose = 0, path = ".", dir = "hitlist",
 prefix = "hitlist", report = NULL)
reportNetworks          (sgi, verbose = 0, path = ".", dir = "networks",
 prefix = "networks", Networks, qv = 0.05,
 withoutgroups = c("pos", "neg"), report = NULL)
reportScreenData        (sgi, type = "data", design = "template",
 do.trafo = TRUE, do.inv.trafo = FALSE, verbose = 0,
   path = ".", dir = "screenplots", prefix = "screenplot",
 png.args = list(width = 960, height = 960),
 pdf.args = list(width = 7, height = 7),
 plotScreen.args = list(ncol = 6L, do.legend = TRUE,
                        fill = c("red", "white", "blue")),
 png.scatter.args = list(width = 400, height = 400),
 pdf.scatter.args = list(width = 7, height = 7), report = NULL)
reportDoublePerturbation(sgi, verbose = 0, path = ".", dir = "doublePerturbation
 prefix = "doublePerturbationPlots", report = NULL,
                         withoutgroups = c("neg", "pos"),
                         png.args = list(width = 500, height = 500),
 pdf.args = list(width = 7, height = 7), ...)
reportMainEffects       (sgi, verbose = 0, path = ".", dir = "maineffects",
 prefix = "maineffects",
 png.args = list(width = 500, height = 500),
 pdf.args = list(width = 7, height = 7),
 plot.args = list(), report = NULL)
reportHeatmap           (sgi, verbose = 0, path = ".", dir = "heatmap",
 prefix = "heatmap",
 png.args = list(width = 1000, height = 1000),
 pdf.args = list(width = 15, height = 15),
 report = NULL, withoutgroups = c("neg", "pos"))
```

Arguments

`sgi` An object of class `RNAinteract`.

`verbose` Either 0 (default, no output), 1 (minimum output), or 2 (outout)

`path` The main path to the HTML report directory.

`dir` A subdirectory where the report is written to.

`prefix` A prefix for each file written in the subdirectory. Using different prefixes, one can write multiple reports in the same directory.

`report` A report object as generated by `startReport`.

`Networks` A boolean array with edges from interaction graphs.

`qv` A cut-off value for the q-values.

`withoutgroups`
Genes annotated with these groups are not plotted in this report.

`type` Any "type" that can be passed to `getData`.

`design` Either "template" (default) or "query"

`do.trafo, do.inv.trafo`
Apply the (inverse) transformation before plotting.

`png.args` A list with entries width and height specifying the width and height of the generated png images.

`pdf.args` A list with entries width and height specifying the width and height of the generated of files.

`plotScreen.args`
Arguments for the screen plots

`png.scatter.args, pdf.scatter.args, plot.args`
Arguments for the scatter plots

`...` Parameters passed to the plotting functions.

Details

Each of these function generates a HTML report. It is added to a HTML frame.

The report object has to be created with startReport beforehand. Multiple report functions can be called afterwards. When all reports are written, the report is finalized and closed by `endReport`.

Value

All functions return a report object as returned by `startReport`.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`, `startReport`, `endReport`

Examples

```
data("sgi")
report = startReport("report")
reportAnnotation(sgi, report = report)
endReport(report)
# browseURL(file.path("report","index.html"))
```

`sgi` *Dataset of class 'RNAinteract'*

Description

Sample object of class `RNAinteract`. The data are real but anonymized. The object contains two replicate screens with three readout channels.

Usage

```
data(sgi)
```

Format

The data contains two screens with three readout channels. The screen is performed on multiwell plates with 8 x 12 wells.

Examples

```
data(sgi)
sgi
```

---

sgisubset *subset of an RNAinteract object.*

Description

A new object of class `RNAinteract` is created that contains a subset of screens and channels.

Usage

```
sgisubset(sgi, screen = getScreenNames(sgi), channel = getChannelNames(sgi))
```

Arguments

`sgi`	An object of class `RNAinteract`.
`screen`	Names of the selected screens.
`channel`	Names of the selected channels.

Details

This function returns a `RNAinteract` object that only contains the selected screens and channels.

Value

An object of class `RNAinteract`.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")
sgi
sgi1 <- sgisubset(sgi, screen = "1")
sgi1
sgi2 <- sgisubset(sgi, channel = "nrCells")
sgi2
```

`sgisubsetQueryDesign` *Subsetting query genes in a RNAinteract object.*

Description

A `RNAinteract` object with a subset of query genes is returned.

Usage

```
sgisubsetQueryDesign(sgi, query.targets = NULL, query.reagents = NULL)
```

Arguments

`sgi` An `RNAinteract` object.

`query.targets`
The query target ids to be selected.

`query.reagents`
The query reagent ids to be selected.

Value

Returns an oject of class `RNAinteract`.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

`startReport` *start and end a RNAinteract report*

Description

`startReport` will open a html page and starts writing an html report for a RNAinteract screen. `endReport` finishes the report and closes the html-file.

Usage

```
startReport(outputpath)
endReport(report)
```

Arguments

`outputpath` the path to the output directory were the report is written to.

`report` An report object as returned by startReport or any report... function.

Details

~~ details ~~

Value

startReport returns an report object. It is handed over to each report-function.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`, `reportAnnotation`, `reportStatistics`, `reportGeneLists`, `reportNetworks`, `reportScreenData`, `reportDoublePerturbation`, `reportMainEffects`

Examples

```
data("sgi")
report <- startReport("report")
reportAnnotation(sgi, report = report)
endReport(report)
# browseURL(file.path("report","index.html"))
```

`summarizeScreens` *summarize screens*

Description

Creates a new object of class `RNAinteract` with one screen. The new screen is the mean of all screens in the input object.

Usage

```
summarizeScreens(sgi, screens, newscreenname = "mean")
```

Arguments

`sgi` An object of class `RNAinteract`.

`screens` The screen names to be summarized.

`newscreenname`
The name of the new summary screen.

Details

If multiple screens with the same layout are stored in the same `RNAinteract` object, then these screens are summarized by averaging to a new screen. The returned object contains one screen.

Value

An object of class `RNAinteract`.

Author(s)

Bernd Fischer

See Also

`RNAinteract-package`

Examples

```
data("sgi")
sgi
sginew <- summarizeScreens(sgi, screens=c("1","2"), newscreenname = "m")
sginew
```

`swaptree` *Swaps a branch of a hclust object.*

Description

Swaps the left and right branch at a specified level of a dendrogram.

Usage

```
swaptree(hc, level)
```

Arguments

`hc`	An hierarchical clustering object as produced by hclust.
`level`	The level to be swapped.

Value

Returns an hclust object.

Author(s)

Bernd Fischer

Index