SIM

April 19, 2009

`RESOURCERER.annotation.to.ID` *Link RESOURCERER annotation file to expression ID*

Description

Get annotation out of the RESOURCERER annotation file and link them to expression data with help of expression ID's

Usage

```
RESOURCERER.annotation.to.ID(data = expr.data, poslist = poslist_expr, col.ID.li
```

Arguments

`data` data.frame with expression data including an expression ID column.

`poslist` data.frame containing the RESOURCERER annotation file

`col.ID.link` numeric value, specifying the column of `data` that contains the ID to link with the `poslist`.

`col.poslist.link`
numeric value, specifying the column of `poslist` that contains the ID to lin k with the `data`.

Details

This function will output the inserted dataset, including the necessary, for integrated.analysis, a nnotation columns: `"CHROMOSOME"`, `"STARTPOS"`and `"Symbol"` out of the inserted RESOURCE RER annotation file `poslist`.

Value

A data.frame is returned, containing a dataset with annotation columns which can be used for integrated.analysis

Author(s)

Marten Boetzer, Melle Sieswerda, Renee x Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

link.metadata

Examples

```
# download expression array annotation from RESOURCERER ftp://occams.dfci.harvard.edu/pub
# it may be necessary to remove the first row, which states the genome build used for map
## Not run: read.an <- read.delim("affy_U133Plus2.txt", sep="\t", header=T)

# get physical mapping columns
## Not run: expr.data <- RESOURCERER_annotation_to_ID(data = read.expr, poslist = read.ar
```

SIM-package *Statistical Integration of Microarrays*

Description

SIM is a statistical model to identify copy number changes that affect the expression of genes within the same chromosomal region. Copy number is considered as the dependent variable and expression as the independent variable. Copy number alterations may span many expression probes and affect them in a possibly subtle but consistent way. Therefore, we test whether copy number is associated with a set of expression levels within a chromosome arm (or mimimal common region) in a random-effect model. Association scores for individual expression levels (z-scores) are also calculated. For more information on the random-effect model, see `?globaltest`.

Each sample should be profiled both on a copy number and on an expression array. The array platforms used for DNA and RNA analysis may be different as long as the probes have mapped to the genome. RESOURCERER can be used to search chromosome and basepair location for expression microarray probes (`http://compbio.dfci.harvard.edu/tgi/cgi-bin/magic/r1.pl`). See `RESOURCERER.annotation.to.ID` on how to insert this information as annotation columns. Alternatively, the chromosome, basepair locations and gene symbol can be extracted from AnnotationData packages available in Bioconductor or generated using the AnnBuilder package.

When copy number data is run as dependent variable, we use `method.adjust="BY"` for multiple testing correction. This method accounts for dependence between measurements and is more conservative than "BH". For details on the multiple testing correction methods see ?p.adjust. We have experienced that a rather low stringency cut-off on the BY-values of 20% allows the detection of associations for data with a low number of samples or a low frequency of abberations. False positives are rarely observed.

Make sure that the array probes are mapped to the same builds of the genome, and that the `chrom.table` used by the `integrated.analysis` is from the same build as well. See `sim.update.chrom.table`.

Details

Package:	SIM
Type:	Package
Version:	1.9.0
Date:	2008-02-06
License:	Open

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

References

R.X. de Menezes, M. Boetzer, M. Sieswerda, G.J.B. van Ommen, J.M. Boer Integrated Statistical analysis to identify associations between DNA copy number and gene expression in microarray data. Submitted.

See Also

assemble.data, integrated.analysis, sim.plot.zscore.heatmap, sim.plot.pvals.on.regio
sim.plot.pvals.on.genome, tabulate.pvals, tabulate.top.dep.features,
tabulate.top.indep.features, impute.nas.by.surrounding, sim.update.chrom.table

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data,ann.dep = colnames(acgh.data)[

#run the integrated analysis
integrated.analysis(samples = samples, input.regions = 8, adjust=FALSE, zscores=TRUE, met

# use functions to plot the results of the integrated analysis

#plot the p-values along the genome
sim.plot.pvals.on.genome(input.regions = 8,adjust.method = "BY",pdf = FALSE, run.name = "

#plot the p-values along the regions
sim.plot.pvals.on.region(input.regions = 8, adjust.method="BY", run.name = "chr8")

#plot the z-scores in an association heatmap
sim.plot.zscore.heatmap(input.regions = 8, significance=0.2, z.threshold=3, show.names.de

#tabulate the p-values per region (prints to screen)
tabulate.pvals(input.regions = 8,adjust.method="BY", bins=c(0.001,0.005,0.01,0.025,0.05,0

#get the top dependent features sorted by p-value
tabulate.top.dep.features(input.regions = 8, adjust.method="BY",run.name = "chr8")

#get the top independent features sorted by mean z-score
tabulate.top.indep.features(input.regions = 8,adjust.method="BY", significance=0.2, sort.
```

acgh.data *Array-comparative genomic hybridization data example*

Description

Copy number log ratios derived from Pollack et al. 2002 PNAS 99:12963-8.

Usage

```
data(acgh.data)
```

Format

A data frame with 4865 observations on the following 45 variables.

describe

Details

If necessary, more details than the description above

Source

reference to a publication or URL from which the data were obtained

References

possibly secondary sources and usages

Examples

```
data(acgh.data)
## maybe str(acgh.data) ; plot(acgh.data) ...
```

assemble.data *Assemble the data to run the integrated analysis*

Description

Assembles the copy number and expression data and annotation.

Usage

```
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(a
```

Arguments

`dep.data` — `data.frame`. The dependent data, along with annotations. Each row should correspond to one feature. The following columns are expected to exist, and the column names should be inserted in the function. `dep.id.`: A unique identifier. `dep.chr.`: The number of the chromosome (chrX=23 and chrY=24). `dep.pos.`: The base pair position, relative to the chromosome. `dep.symb.`: Gene symbol (optional). The data will be sorted on `Abs.start`, generated by chr*10e9+basepair.

`indep.data` — `data.frame`. The independent data, along with annotations. Each row should correspond to one feature. The following columns are expected to exist, and the column names should be inserted in the function. `indep.id.`: A unique identifier. `indep.chr.`: The number of the chromosome (chrX=23 and chrY=24). `indep.pos.`: The base pair position, relative to the chromosome. `indep.symb.`: Gene symbol (optional). The data will be sorted on `Abs.start`, generated by chr*10e9+basepair.

`ann.dep` — `vector` with either the names of the columns or the column numbers in the dependent data that contain the annotation.

`ann.indep` — `vector` with either the names of the columns or the column numbers in the independent data that contain the annotation.

`dep.id` — `vector` with the column name in the dependent data that contains the ID. Will be used in the `sim.plot.zscore.heatmap` function. Empty ID's will be substituted by NA.

`dep.chr` — `vector` with column name in the dependent data that contains the chromosome numbers.

`dep.pos` — `vector` with the column name in the dependent data that contains the position on the chromosome in bases.

`dep.symb` — Optional, either F(alse) or a single vector with the column name in the dependent data that contains the Symbols. Will be used in `sim.plot.zscore.heatmap` as label.

`indep.id` — `vector` with the column name in the independent data that contains the ID. Will be used in the `sim.plot.zscore.heatmap` function. Empty ID's will be substituted by NA.

`indep.chr` — `vector` with the column name in the independent data that contains the chromosome numbers.

`indep.pos` — `vector` with the column name in the independent data that contains the position on the chromosome in bases.

`indep.symb` — Optional, either F(alse) or a vector with the column name in the dependent data that contains the Symbols. Will be used in `sim.plot.zscore.heatmap` as label.

`overwrite` — `Boolean`, indicate when a `run.name` is already present, the results can be overwritten.

`run.name` — Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

Value

No values are returned. Instead, the datasets and annotation columns are stored in seperate files in the `data` folder in the directory specified in `run.name`. If the `assemble.data` function has run succesfully, the `integrated.analysis` function can be performed.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.region`, `sim.plot.pvals.on.genome`, `tabulate.pvals`, `tabulate.top.dep.features`, `tabulate.top.indep.features`, `impute.nas.by.surrounding`, `sim.update.chrom.table`

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#read the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(acgh.data)
```

`chrom.table` *Table with chromosome arm positions*

Description

A table indicating the base positions of the beginning and end of chromosome arms. Currently based on the UCSC March 2006 / NCBI 36 build of the human genome.

Usage

```
data(chrom.table)
```

Format

A data frame with 48 observations on the following 6 variables.

describe

Details

If necessary, more details than the description above

Source

reference to a publication or URL from which the data were obtained

References

possibly secondary sources and usages

Examples

```
data(chrom.table)
## maybe str(chrom.table) ; plot(chrom.table) ...
```

`expr.data` *Expression data example*

Description

Expression log ratios derived from Pollack et al. 2002 PNAS 99:12963-8.

Usage

```
data(expr.data)
```

Format

A data frame with 4865 observations on the following 45 variables.

describe

Details

If necessary, more details than the description above

Source

reference to a publication or URL from which the data were obtained

References

possibly secondary sources and usages

Examples

```
data(expr.data)
## maybe str(expr.data) ; plot(expr.data) ...
```

`impute.nas.by.surrounding` *Impute NA's in array-CGH data*

Description

Replace an NA by the median of the surrounding features in the same sample.

Usage

```
impute.nas.by.surrounding(dataset, window.size = 5)
```

Arguments

`dataset` data.frame with the dataset to replace the NA's by the medians of the surrounding features.

`window.size` numeric value, specifying of how many features around the NA the median should be taken.

Details

This function can be used when the dependent dataset in the integrated.analysis function is array-CGH data and contains probes that have an NA. To avoid loosing data by throwing away the probes with NA's, the impute.nas.by.surrounding function can be used which simply takes the median of the probes around an NA. The number of probes used for the imputatin is chosen by giving a value for window.size. This script takes quite long to run!

Value

A data.frame is returned, containing the inserted `dataset` without NA's, which are medianed.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.r` `sim.plot.pvals.on.genome`, `tabulate.pvals`, `tabulate.top.dep.features`, `tabulate.top.indep.features`, `sim.update.chrom.table`

Examples

```
## Not run: cgh.imp <- impute.nas.by.surrounding(cgh)
```

`integrated.analysis` *Integrated analysis of expression and copy number microarray data*

Description

Runs the Integrated Analysis to test for associations between DNA copy number measurements and gene expression measurements on the same set of samples.

Usage

```
integrated.analysis(samples, input.regions="all chrs", adjust=FALSE, zscores=FAL
  method=c("auto", "asymptotic", "permutations", "gamma"), run.name=NULL)
```

Arguments

`samples` `vector` with either the names of the columns in the dependent and independent data corresponding to the samples, or a numerical vector containing the column numbers to include in the analysis, e.g. 5:10 means columns 5 till 10. Make sure that both datasets have the same number of samples with the same column names!

`input.regions`
`vector` indicating the regions to be analyzed. Can be defined in four ways: `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`,`"all chrs auto"`,`"all arms"`,`"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own vector of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector`: `c(1, 2, 3). 3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q").4) subregions of a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3)`. See `details` for more information.

`adjust` Confounders for which the integration-test must be adjusted, such as tumor type, location, gender, etc. Either a formula with a factor of a vector with names with the same length as samples or FALSE. A formula with a vector can e.g. be: `Y~factor(subtype)` where subtype is a vector with the same length as samples with names like: subtype = c("tumor","tumor", "normal","normal", etc...) See `?globaltest` for more information.

`zscores` `Boolean`, indicates whether the z-scores are calculated (takes longer time to run). If z-scores=FALSE, only p-values are calculated.

`method` The method for calculation of the p-values. Use `method = "asymptotic"` for the full asymptotic distribution of the test statistic, `method = "gamma"` for the gamma (= scaled chi-squared) approximation to that distribution and `method = "permutations"` for a permutation p-value. The recommended default: `method = "auto"` chooses the permutations method if the number of possible permutations does not exceed 10,000 and the asymptotic otherwise. See `?globaltest` for more information.

`run.name` Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

Details

The Integrated Analysis is a regression of the independent data on the dependent features. In most cases, the dependent data will be the copy number measurements from array-CGH and the independent data the expression array values. The regression itself is done using the `globaltest`, which means that the genes in a region (e.g. a chromosome arm) are tested as a gene set. The individual associations between each copy number probe and each expression probe are calculated as z-scores (standardized influences, see `?globaltest`).

This function splits the datasets into separate sets for each region (as specified by the `input.regions`) and runs the analysis for each region separately.

When running the Integrated Analysis for a predefined input region, like "all arms" and "all chrs", output can be obtained for all input regions, as well as subsets of it. But note that the genomic unit must be the same: if integrated.analysis was run using chromosomes as units, any of the functions and plots must also use chromosomes as units, and not chromosome arms. Similarly, if integrated analysis was run using chromosome arms as units, these units must also be used to produce plots and outputs. For example if the input.regions = "all arms" was used, p-value plots (see sim.plot.pvals.on.region can be produced by inserting the input.regions = "all arms", but also for instance "1p" or "20q". However, to produce a plot of the whole chromosome, for example chromosome 1, the integrated should be re-run with input.region=1. The same goes for "all chrs": p-value plots etc. can be produced for chromosome 1,2 and so on... but to produce plots for an arm, the integrated.analysis should be re-run for that region. This also goes for subregions of the chromosome like "chr1_1-1000000".

Value

No values are returned. Instead, the results of the analysis are stored in the subdirectories of the directory specified in run.name. E.g. the z-score matrices are saved in subfolder intermediate.data. The following functions can be used to visualize the data:

1) sim.plot.zscore.heatmap (pdf, only possible when zscores=TRUE)

2) sim.plot.pvals.on.region (pdf)

3) sim.plot.pvals.on.genome (pdf)

Other functions can be used to tabulate the results:

1) tabulate.pvals (data.frame)

2) tabulate.top.dep.features (txt)

3) tabulate.top.indep.features (txt, only possible when zscores=TRUE

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

References

1 Goeman JJ, van de Geer SA, de Kort F, van Houwelingen HC. A global test for groups of genes: testing association with a clinical outcome. Bioinformatics. 2004; 20:93-109

See Also

SIM, assemble.data, sim.plot.zscore.heatmap, sim.plot.pvals.on.region, sim.plot.pvals.on.genome, tabulate.pvals, tabulate.top.dep.features, tabulate.top.indep.features, impute.nas.by.surrounding, sim.update.chrom.table,glob

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
```

```
assemble.data(dep.data=acgh.data, indep.data=expr.data, ann.dep=colnames(acgh.data)[1:4],

#run the integrated analysis
integrated.analysis(samples=samples, input.regions=c(8), adjust=FALSE, zscores=TRUE, meth
```

link.metadata *Link a metadata annotation file to expression ID*

Description

Get annotation out of a AnnotationData package and link them to the expression data using the expression probe ID's

Usage

```
link.metadata(data = expr.data, col.ID.link = 1, chr = as.list(hgu133plus2CHR),
```

Arguments

data	`data.frame` with expression data including an expression probe ID column.
col.ID.link	numeric value, specifying the column of `data` that contains the ID to link with the `poslist`.
chr	`list` specifying the metadata annotation of the chromosome location on the genome.
chrloc	`list` specifying the metadata annotation of the location of the probe on the chromosome.
symbol	`list` specifying the metadata annotation of the symbol corresponding to the probe.

Details

Often, the annotation for expression array probes lack chromosome position information. Therefore, this function adds the two required columns to run the `integrated.analysis`: "CHROMOSOME" and "STARTPOS". In addition, the optional column, "Symbol" is added.

Value

A `data.frame` is returned, containing a dataset with annotation columns which can be used for `integrated.analysis`.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee x Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

RESOURCERER.annotation.to.ID

Examples

```
# first download and install the AnnotationData package for your expression array platfor
# for example
## Not run: library(hgu133plus2)
## Not run:
expr.data <- link.metadata(data, col.ID.link = 1, chr = as.list(hgu133plus2CHR),
chrloc = as.list(hgu133plus2CHRLOC), symbol = as.list(hgu133plus2SYMBOL))
## End(Not run)
```

`samples` *Samples for example data*

Description

Vector of sample names corresponding to the column headers containing the data in both the copy number (`acgh.data`) and expression (`expr.data`) example datasets.

Usage

```
data(samples)
```

Format

The format is: chr [1:41] "BT474" "MCF7" "NORWAY.10" "NORWAY.100" "NORWAY.101" ...

Details

If necessary, more details than the description above

Source

reference to a publication or URL from which the data were obtained

References

possibly secondary sources and usages

Examples

```
data(samples)
## maybe str(samples) ; plot(samples) ...
```

sim.plot.pvals.on.genome

Plot the p-values in whole genome overview

Description

Generates a plot of the analyzed dependent data probe positions and their significance on all chromosomes.

Usage

```
sim.plot.pvals.on.genome(input.regions="all chrs", adjust.method=c("BY", "BH", "
```

Arguments

`input.regions`
: `vector` with analyzed regions for which to produce the graphs. Can be defined in four ways: `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`, `"all chrs auto"`, `"all arms"`, `"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own vector of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector`: `c(1, 2, 3). 3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q"). 4) subregions of a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3).` For more information see the `details` section of integrated.analysis.

`adjust.method`
: Method used to adjust the p-values for multiple testing. Either `"BY"` (recommended when copy number is used as dependent data), `"BH"` or `"raw"`. Defaults to "BY". See SIM for more information about adjusting p-values.

`pdf`
: Logical. Indicate whether to generate a pdf of the plot in the run.name directory or plot on screen.

`run.name`
: Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

`...`
: Arguments to be passed to methods, such as graphical parameters (see par).

Details

The orange triangles indicate the start and end of the analyzed regions. The purple dot indicates the cent romere.

Value

No values are returned. The results are stored in the folder `run.name` as pdf.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

SIM, assemble.data, integrated.analysis, sim.plot.zscore.heatmap, sim.plot.pvals.on.r
tabulate.pvals, tabulate.top.dep.features, tabulate.top.indep.features,
impute.nas.by.surrounding, sim.update.chrom.table

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data=acgh.data, indep.data=expr.data, ann.dep=colnames(acgh.data)[1:4],

#run the integrated analysis
integrated.analysis(samples=samples, input.regions=c(8), adjust=FALSE, zscores=TRUE, meth

#plot the p-values along the genome
sim.plot.pvals.on.genome(input.regions=c(8), adjust.method="BY", pdf=FALSE, run.name="chr
```

---

sim.plot.pvals.on.region

P-value histograms and p-values along the genome per region

Description

Generates two plots of the p-values for an analyzed region. The first plot contains the distribution of the raw p-values and ranked plots of the raw and adjusted p-values. The second plot contains the p-values along the genome of analyzed input.regions.

Usage

```
sim.plot.pvals.on.region(input.regions = c("all chrs"), adjust.method = c("BY",
```

Arguments

input.regions
: vector with analyzed regions for which to produce the graphs. Can be defined in four ways: 1) predefined input region: insert a predefined input region, choices are: "all chrs", "all chrs auto", "all arms", "all arms auto" In the predefined regions "all arms" and "all arms auto" the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own vector of arms. 2) whole chromosome(s): insert a single chromosome or a list of chromosomes as a vector: c(1, 2, 3). 3) chromosome arms: insert a single chromosome arm or a list of chromosome arms like c("1q", "2p", "2q"). 4) subregions of

`a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3)`. For more information see the `details` section of `integrated.analysis`.

`adjust.method`
Method used to adjust the p-values for multiple testing. Either `"BY"` (recommended when copy number is used as dependent data), `"BH"` or `"raw"`. Defaults to "BY". See `SIM` for more information about adjusti ng p-values.

`run.name` Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

`...` Arguments to be passed to methods, such as graphical parameters (see par).

Details

This function returns a pdf containing the p-value plots. The second plot contains the multiple testing corrected p-values plotted along the chromosome (arm). On the x-axis, the start positions of the dependent features are displayed. On the y-axis, the p-value levels are displayed. Two dotted lines indicate p-value levels 0.2 and 0.1. In general, p-values below 0.2 are said to be "significant".

Value

No values are returned. The results are stored in a subdirectory of `run.name` as pdf.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.g`
`tabulate.pvals`, `tabulate.top.dep.features`, `tabulate.top.indep.features`,
`impute.nas.by.surrounding`, `sim.update.chrom.table`

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(acgh.data)

#run the integrated analysis
integrated.analysis(samples = samples, input.regions = c(8), adjust=FALSE, zscores=TRUE,

# use functions to plot the results of the integrated analysis

#plot the p-values along the region
sim.plot.pvals.on.region(input.regions = c(8), adjust.method="BY", run.name = "chr8")
```

`sim.plot.zscore.heatmap` *Association heatmap from z-scores*

Description

Produces an association heatmap that shows the association (standardized influence) of each independent feature (expression measurement) with each dependent feature (copy number measurement). A p-value bar on the left indicates test signficance. A color bar on top indicates genes with mean z-scores across the signficant copy number probes above a set threshold. A summary of the copy number data helps to identify what copy number alterations are present in a region of association with expression. Positive association can mean copy number gain and increased expression, or deletion and decreased expression. The heatmaps can also be used in an exploratory analysis, looking for very local effects of copy number changes (usually small amplifications) on gene expression, that do not lead to a significant test result.

Usage

```
sim.plot.zscore.heatmap(input.regions = "all chrs", significance = 0.2, z.thresh
```

Arguments

`input.regions`
: `vector` indicating the regions to be analyzed. Can be defined in four ways: `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`, `"all chrs auto"`, `"all arms"`, `"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own `vector` of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector`: `c(1, 2, 3). 3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q"). 4) subregions of a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3).` See `details` for more information.

`significance`
: Threshold to select the significant dependent features. Only these features are used to calculate the mean z-scores per independent feature (expression probe).

`z.threshold`
: Threshold to display a green or red bar in the color bar on top of the heatmap for independent features with mean z-scores above `z.threshold` (high positive association) or below `-z.threshold` (high negative association).

`show.names.indep`
: `Boolean`. If set to TRUE, displays the names (`indep.id` and `in dep.symb` entered in the `assemble.data`) of the independent features with mean z-scores above or below the `z.threshold` in the heatmap.

`show.names.dep`
: `Boolean`. If set to TRUE, displays the names (`dep.id` and `dep.sy mb` entered in the `assemble.data`) of the `significant` dependent features in the heatmap.

adjust.method
: Method used to adjust the p-values for multiple testing. Either "BY" (recommended when copy number is used as dependent data), "BH" or "raw". See SIM for more information about adjusting p-values. Defaults to "BY".

scale
: Vector specifying the color scale in the heatmap. If scale="auto", the maximum and minimum value of all z-scores will be calculated and set as the limits for all analyzed regions. Another option is to define a custom scale, e.g. scale = c(-5,5).

plot.method
: Summary plot of copy number data in left panel. Either "clac", "smooth","heatmap", or "none". Should only be used when the dep.data is array-CGH. The "clac" plot is a consensus of the aberration frequencies across all samples. CLAC requires at least three normal/diploid arrays. For more details see ?clac.preparenormal The "smooth" plot smoothes the copy number log ratios per sample, see ?quantsmooth for more details. The "heatmap" method produces an aCGH heatmap where green indicates gain, and red loss. The scale of the aCGH heatmap is automatically set to the min and max of the aCGH measurements of the analyzed regions. Default is plot.method = "none", no additional plot will be drawn.

Normal.data
: vector, required for plot.method = "clac", indicating least three normal samples in the dependent data. Insert the column names of the samples that are normal e.g. for the first three samples: Normal.data = 1:3. If no normal samples are available, use Normal.data = FALSE. Then Normal.data are generated by calculating probe medians of three subsets of the dependent data.

windowsize
: Numeric value, specifying the window size to carry out the average smooth for plot.method="clac". For more details see ?clac.preparenormal.R.

lambda
: Numeric value, specifying the quantile smoothing parameter for plot.method="smooth". See ?quantsmooth and references for more information.

subtype
: This variable must be a vector with the same length as samples or FALSE. The vector will be transformed to a factor and the levels of this will be coloured according to their subtype. When subtype=FALSE, all the samples will be coloured black.

acgh.heatmap.scale
: Vector specifiing the color scale in the aCGH heatmap. If scale="auto", the maximum and minimum value of all aCGH values will be calculated and set as the limits for all analyzed regions. Another option is to define a custom scale, e.g. scale = c(-5,5).

pdf
: Logical. Indicate whether to generate a pdf of the plots in the heatmap_zscores subdirectory or plot to screen.

run.name
: Name of the analysis. The results will be stored in a folder with this name in the current working directory (use getwd() to print the current working directory). If the run.name = NULL, the default folder "analysis_results" will be generated.

...
: additional arguments passed on to image

Details

The sim.plot.zscore.heatmap function can only run after the integrated.analysis is run with zscores = TRUE.

The results are returned as a single-page pdf containing an association heatmap of the regions listed in input.regions. For high-density arrays large files will be produced, both demanding more memory available from your computer to produce them as well as being heavier to open on screen. To avoid this, analyze chromosome arms as units instead of chromosomes, both here and in `input.regions = "all arms"`.

The heatmap contains the z-scores generated by the function `integrated.analysis` with `zscores=T`. The dependent features are plotted from bottom to top, the independent features from left to right. Positive associations are shown in green, negative associations in red (color scale on the right). At the left side of the heatmap a color bar represents the multiple testing corrected p-values of the probes in the dependent data (copy number), also with a color legend. Dependening on which `plot.method` is used, a summary of copy number changes is shown on the left. At the top of the heatmap is a color bar corresponding to the mean z-scores of the independent features (expression data) that are above or below the `z.threshold`. If `show.names.indep` is set to TRUE, labels will be drawn for the probes with mean z-scores greater than `z.threshold` or lower than `-z.threshold` at the bottom of the heatmap. If `show.names.dep` is set to TRUE, labels will be drawn for the significant dependent probes lower than `significance` to the right of the heatmap.

Value

No values are returned. The results are stored in a subdirectory of `run.name` as pdf.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

References

1 Eilers PH, de Menezes RX. Quantile smoothing of array CGH data. Bioinformatics. 2005 Apr 1;21(7):1146-53.

2 Wang P, Kim Y, Pollack J, Narasimhan B, Tibshirani R. A method for calling gains and losses in array CGH data. Biostatistics. 2005; 6:45-58.

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.pvals.on.region`, `sim.plot.pvals.on.`
`tabulate.pvals`, `tabulate.top.dep.features`, `tabulate.top.indep.features`, `impute.nas.by.surrounding`, `sim.update.chrom.table`,`image.plot`,`maPalette`

Examples

```
#load the datasets and the samples to run the integration for
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(acgh.data)

#run the integrated analysis
integrated.analysis(samples = samples, input.regions = 8, adjust=FALSE, zscores=TRUE, met

# use functions to plot the results of the integrated analysis
```

```
#plot the zscores in a heatmap
sim.plot.zscore.heatmap(input.regions = 8, significance=0.2, z.threshold=3, show.names.de
```

`tabulate.pvals` *Sums significant p-values for the analyzed regions*

Description

Generates a `data.frame` with the signicance of p-values in the analyzed regions, dividing them into bins.

Usage

```
tabulate.pvals(input.regions = "all chrs", adjust.method = "BY", bins = c(0.001,
```

Arguments

`input.regions`
: `vector` with analyzed regions for which to produce the table. Can be defined in four ways:

  `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`, `"all chrs auto"`, `"all arms"`, `"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own `vector` of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector`: `c(1, 2, 3)`. `3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q")`. `4) subregions of a chromosome:` insert a chromosome number followed by the start and end position lik e `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3)`. See `details` for more information.

`adjust.method`
: Method used to adjust the p-values for multiple testing. Either `"BY"` (recommended when copy number is used as dependent data), `"BH"` or `"raw"`. See `SIM` for more information about adjusting p-values.

`bins`
: `vector` of significance thresholds. This function will calculate the number of features having a p-value lower than the bin.

`significance.idx`
: Index of "bins" to use when computing the percentage of significant p-values. Defaults to 8 (i.e. the first entry in "bins"), in this case 0.20.

`order.by`
: Column used for sorting the table. Defaults to "%" (i.e. the percentage of significant p-va lues).

`decreasing`
: Direction used for sorting. Defaults to TRUE (i.e. highest values on top).

`run.name`
: Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

Value

Returns a `data.frame`. Each row corresponds to a chromosome and has as many entries as entries in bins, plus 1. Each entry contains the number of p-values that is smaller or equal to the corresponding entry in bins.

The last entry holds the percentage of p-values that is smaller than or equal to the bin identified by `significance.idx`.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.r` `sim.plot.pvals.on.genome`, `tabulate.top.dep.features`, `tabulate.top.indep.features`, `impute.nas.by.surrounding`, `sim.update.chrom.table`

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = 1:4,ann.indep = 1:4

#run the integrated analysis
integrated.analysis(samples = samples, input.regions = 8, adjust = FALSE, zscores=TRUE, m

#tabulate the p-values per region
tabulate.pvals(input.regions = 8,adjust.method="BY", bins=c(0.001,0.005,0.01,0.025,0.05,0
```

`tabulate.top.dep.features` *Lists the p-values for the dependent features*

Description

Lists the integrated analysis p-values for the dependent features in the analyzed regions, together with the available annotation.

Usage

```
tabulate.top.dep.features(input.regions = "all chrs", adjust.method = c("BY", "B
```

Arguments

`input.regions`
: `vector` indicating the regions to be analyzed. Can be defined in four w ays: `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`, `"all chrs auto"`, `"all arms"`, `"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own `vector` of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector`: `c(1, 2, 3)`. `3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q")`. `4) subregions of a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3)`. See `details` for more information.

`adjust.method`
: Method used to adjust the p-values for multiple testing. Either `"BY"` (recommended when copy number is used as dependent data), `"BH"` or `"raw"`. Defaults to "BY". See `SIM` for more information about adjustin g p-values.

`run.name`
: Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

Details

Output is a .txt file containing a table with sorted integrated analysis p-values of the dependent features. It includes the `ann.dep` columns that were read in the `assemble.data` function.

Value

No values are returned. The results are stored in a subdirectory of `run.name` as txt.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.r`
`sim.plot.pvals.on.genome`, `tabulate.pvals`, `tabulate.top.indep.features`,
`impute.nas.by.surrounding`, `sim.update.chrom.table`

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(acgh.data)
#run the integrated analysis
integrated.analysis(samples = samples, input.regions = 8, adjust=FALSE, zscores=TRUE, met

#get the top dependent features with lowest p-value
tabulate.top.dep.features(input.regions = 8, adjust.method="BY",run.name = "chr8")
```

---

tabulate.top.indep.features

Lists the mean z-scores for the independent features

Description

Lists the mean z-scores for independent features in the analyzed regions, calculated across the significant dependent features. Gives insight in the expression levels most strongly associated with copy number changes.

Usage

```
tabulate.top.indep.features(input.regions = "all chrs", adjust.method = c("BY",
```

Arguments

`input.regions`
: `vector` indicating the regions to be analyzed. Can be defined in four ways: `1) predefined input region:` insert a predefined input region, choices are: `"all chrs"`, `"all chrs auto"`, `"all arms"`, `"all arms auto"` In the predefined regions `"all arms"` and `"all arms auto"` the arms 13p, 14p, 15p, 21p and 22p are left out, because in most studies there are no or few probes in these regions. To include them, just make your own `vector` of arms. `2) whole chromosome(s):` insert a single chromosome or a list of chromosomes as a `vector` `c(1, 2, 3)`. `3) chromosome arms:` insert a single chromosome arm or a list of chromosome arms like `c("1q", "2p", "2q")`. `4) subregions of a chromosome:` insert a chromosome number followed by the start and end position like `c("chr1_1-1000000")` These regions can also be combined, e.g. `c("chr1_1-1000000","2q", 3)`. See `details` for more information.

`adjust.method`
: Method used to adjust the p-values for multiple testing. Either `"BY"` (recommended when copy number is used as dependent data), `"BH"` or `"raw"`. Defaults to "BY". See `SIM` for more information about adjustin g p-values.

`significance`
: threshold used to select the significant dependent features. Only the z-scores with these features are used to calculate the mean z-scores across the independent features.

`sort.order`
: Indicates how the z-scores are sorted, either `"positive"` or `"negative"`.

`run.name`
: Name of the analysis. The results will be stored in a folder with this name in the current working directory (use `getwd()` to print the current working directory). If the `run.name = NULL`, the default folder `"analysis_results"` will be generated.

Details

`tabulate.top.indep.features` can only be run after `integrated.analysis` with `zscores=T`.

Output is a .txt file containing a table with the mean z-scores of all independent features per analyzed region. It includes the `ann.indep` columns that were read in the `assemble.data` function.

Depending on the argument "adjust.method", the p-values are first corrected for multiple testing. Next, th e z-scores are filtered to include only those entries that correspond to significant (p-value < "significa nce") dependent features to calculate the mean z-scores.

Value

No values are returned. The results are stored in a subdirectory of `run.name` as pdf.

Author(s)

Marten Boetzer, Melle Sieswerda, Renee X. de Menezes ⟨R.X.Menezes@lumc.nl⟩

See Also

`SIM`, `assemble.data`, `integrated.analysis`, `sim.plot.zscore.heatmap`, `sim.plot.pvals.on.r` `sim.plot.pvals.on.genome`, `tabulate.pvals`, `tabulate.top.dep.features`, `impute.nas.by.surrounding`, `sim.update.chrom.table`

Examples

```
#load the datasets and the samples to run the integrated analysis
data(expr.data)
data(acgh.data)
data(samples)

#assemble the data
assemble.data(dep.data = acgh.data, indep.data = expr.data, ann.dep = colnames(acgh.data)

#run the integrated analysis
integrated.analysis(samples = samples, input.regions = 8, adjust=FALSE, zscores=TRUE, met

#get the highest associated independent features
tabulate.top.indep.features(input.regions = 8, adjust.method="BY", significance=0.2,sort.
```

Index