Package 'msarc'

January 27, 2015

Version 1.4.5

License Artistic-2.0

Date 2015-01-12

Title Draw Diagrams (mis)Representing the Results of Mass Spec Experiments

Author Gord Brown, Hisham Mohammed

Maintainer Gord Brown <gdbzork@gmail.com>

Depends R (>= 3.1.0)

Imports AnnotationDbi, gplots, XLConnect, wordcloud, RColorBrewer

Suggests GO.db, org.Hs.eg.db, org.Mm.eg.db

Description The output of an affinity-purification mass spectrometry experiment is typically a list of proteins that were observed in the experiment, identified by UniProt identifiers (http://www.uniprot.org/). This package takes as input a list of UniProt identifiers, and the associated Mascot scores from the experiment (which indicate the likelihood that the protein has been correctly identified), clusters them by gene ontology category (http://geneontology.org/), then draws diagrams showing the results in hierarchical clusters by category, with lines for individual proteins representing the associated Mascot score. The results are in general not publication-ready, but will rather require editing via a graphics editor that can interpret SVG (scalable vector graphics) format. As an alternative representation, the package will also generate tag clouds based on the Mascot scores.

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-27 16:55:53

R topics documented:

msarc-package 2
msarc 3

msarc.copyColors 4
msarc.findGOterms 5
msarc.loadMS 6
msarc.loadTerms 7
msarc.plotCircle 8
msarc.plotHeatmap 9
msarc.plotSVG 10
msarc.subtract 11
msarc.tagCloud 12
print.msarc 13
sample_df 14
sample_initial 14

Index **16**

`msarc-package` *msarc: Mass Spectrometry Experiment Plots*

Description

Plots the results of mass spectrometry experiments in a circular format. Lines represent identified proteins, with the length of the line indicating the relevant score, typically the likelihood that the ID is correct, or change relative to some reference. Proteins are clustered by GO categories.

Details

There are several steps to generating mass spec plots:

1. Load the mass spec result file into R;
2. Subtract the proteins that also occur in the control sample;
3. Cluster the proteins into GO categories;
4. Filter the list of categories to be shown;
5. Plot the results.

These steps are largely independent; any one may be replaced with a different function, with some work. For example, the default clustering is according to Gene Ontology terms, but another clustering is also fine. Similarly, although the package is designed for mass spec experiments, it could in principle be used with other sorts of data, provided that it is based on Uniprot IDs, corresponding protein symbols (e.g. CTCF), and some sort of score.

The output format is svg (scalable vector graphics), suitable for editing with Adobe Illustrator or similar.

Please be aware that this program does *not* produce finished, polished figures. They usually need some editing to look right. Also, a few iterations are usually needed to decide exactly which GO terms are most meaningful (at the step where you manually select the set of interest).

Caveat: These plots mis-represent the clustering into GO categories. Most proteins are in more than one category; most GO categories have more than one parent. These plots do not reflect this, but instead show any given protein in only one category (the one with fewest members in this plot), and any given GO category under only one parent (the one with fewest...).

Author(s)

Gord Brown, Hisham Mohammed

Maintainer: Gord Brown <gordon.brown@cruk.cam.ac.uk>

References

This package was inspired in part by the Circos package http://circos.ca/, by Martin Krzywinski (*Martin I Krzywinski, Jacqueline E Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra, "Circos: An information aesthetic for comparative genomics". Genome Research 19 (2000), pages 1639–1645*).

---

msarc *Create msarc Object From Data Frame*

Description

Create an `msarc` object from a data frame.

Usage

```
msarc(df)
```

Arguments

`df`	a data frame with three columns, in this order: Uniprot ID, gene symbol, and score. Column names will be ignored.

Details

This function simply loads the necessary three columns into an `msarc` object and sets some default configuration values.

Value

an `msarc` object corresponding to the data frame.

Author(s)

Gord Brown

Examples

```
# Create an msarc object from a data frame
data(sample_df,package='msarc')
msarc_raw <- msarc(sample_df)
```

`msarc.copyColors` *Copy* `msarc` *Color Palette*

Description

If two or more `msarc` diagrams are presented, it may be helpful if GO categories that occur in multiple diagrams are the same color. This functions allow the copying of the color palette from one `msarc` object to another, so that when the second is plotted, it will resemble the first.

Usage

```
msarc.copyColors(target,source)
```

Arguments

`target`	an `msarc` object to load the color palette into.
`source`	an `msarc` object from which to get a color palette.

Value

the modified target `msarc` object.

Note

`msarc.copyColors` can only be applied to source `msarc` objects for which `msarc.plotSVG` has already been run.

Author(s)

Gord Brown

Examples

```
data(sample_initial,package="msarc")
data(sample_complete,package="msarc")
sample_initial <- msarc.copyColors(sample_initial,sample_complete)
```

`msarc.findGOterms` *Map UniProt Ids to GO Terms*

Description

Maps UniProt accession ids to the (multiple) GO terms they are associated with, and adds the map to the `msarc` object. Any given accession will most likely be associated with multiple GO terms.

Usage

```
msarc.findGOterms(msarc,eg2uniprot=org.Hs.eg.db::org.Hs.egUNIPROT,
                  eg2go=org.Hs.eg.db::org.Hs.egGO,minCount=10)
```

Arguments

`msarc`	the `msarc` object to add GO terms to.
`eg2uniprot`	an `AnnDbBimap` mapping Entrez gene IDs to Uniprot IDs.
`eg2go`	an `AnnDbBimap` mapping Entrez gene IDs to GO categories.
`minCount`	GO categories with fewer than `minCount` members in the UniProt IDs associated with this object will be discarded from the set of "interesting" categories.

Details

The `eg2uniprot` and `eg2go` tables can be found in the annotation packages for various species. For example, the human and mouse packages are `org.Hs.eg.db` and `org.Mm.eg.db`, respectively.

This function is a bit slow.

Value

the modified `msarc` object.

Note

It is normal if a few UniProt accessions cannot be mapped to Entrez IDs, or from there to GO terms. The unmapped accessions will be listed after the mapping is complete, with the message "missing UniProt IDs: ...".

Author(s)

Gord Brown

References

The Gene Ontology Consortium, "Gene Ontology: tool for the unification of biology". Nature Genetics 25:25-29 (2000)

Examples

```
# Load a data set from a mass spec experiment
data(sample_subtracted,package="msarc")

# run the mapping stage
## Not run: sample_go <- msarc.findGOterms(sample_subtracted)
```

`msarc.loadMS` *Create msarc Object From Mass Spec Data*

Description

Create an `msarc` object from a mass spectrometry experiment spreadsheet. The spreadsheet must be saved in tab-separated text format.

Usage

```
msarc.loadMS(file, onlyDiff = T, config=list())
```

Arguments

`file`	the name of the file to load.
`onlyDiff`	If the experiment is a SILAC experiment, and `onlyDiff` is true, only proteins for which both a heavy and a light version was seen will be loaded. If false, all proteins will be included. It is usually best to leave this parameter `TRUE`.
`config`	If the user needs to override column headings in the input file, a named list may be used to specify the column headings. See below for details.

Details

The input file will be parsed according to the following rules:

1. The first line is taken to be column headers.
2. Lines beginning with white space are skipped. (In the output of the mass spec software this is based on, these lines are the descriptions of the individual peptides identified; we only keep the lines describing the whole proteins.)
3. Based on the column headers, three columns are retained: the UniProt ID, the gene symbol and the score. The default column heads are "Accession", "Description" and "Score". The "Description" column is scanned for a pattern like "- [XXX_YYYY]"; the "XXX" is interpreted as the gene symbol.

In addition, if the file has a column named "Heavy/Light", it is taken to be a SILAC experiment; rows without a value in this column are discarded unless `onlyDiff` is `FALSE`.

Column names may be overridden via the configuration list. The defaults are as above; they may be overridden by setting these names in the config list: "accessionCol", "descriptionCol", "scoreCol", and "heavyCol".

Value

an msarc object corresponding to the file.

Author(s)

Gord Brown

Examples

```
# Load a data set from a mass spec experiment
p <- system.file("extdata",package="msarc")
samp_fn <- file.path(p,"sample_msdata.txt.gz")
sample <- msarc.loadMS(samp_fn)

# Load a data set, with a configuration list
samp_fn <- file.path(p,"sample_colnames.txt.gz")
sample <- msarc.loadMS(samp_fn,config=list("accessionCol"="Uniprot_ID"))
```

---

msarc.loadTerms *Manipulate GO Term Lists*

Description

Removing uninteresting GO terms from the list of terms relevant to a set of UniProt accessions must be done manually. msarc.saveTerms and msarc.loadTerms allow the user to save the list of GO terms from a given msarc object to a file for manual filtering, then re-load the filtered file. msarc.getTerms and msarc.filterTerms allow the user to retrieve the list of GO terms as a data frame, filter it manually, and update the msarc object with the filtered list.

Usage

```
msarc.saveTerms(msarc, filename="go_terms.txt")
msarc.loadTerms(msarc, filename="go_terms.txt")
msarc.getTerms(msarc)
msarc.filterTerms(msarc, keepers)
```

Arguments

msarc	an msarc object.
filename	the name of a file to load from or save to.
keepers	a filtered data frame from msarc.getTerms, or a vector of GO IDs to keep.

Details

The file generated by `msarc.saveTerms` must be modified only by having complete lines containing unwanted GO terms deleted.

`msarc.getTerms` returns the list of GO terms as a data frame, with columns for the GO ID, name of the category, and number of proteins in this category. The user can filter this list manually, then use `msarc.filterTerms` to indicate which categories should be represented in the final plot. `msarc.filterTerms` is passed an `msarc` object, and either a filtered data frame from `msarc.getTerms` or a vector of Uniprot IDs.

Value

`msarc.loadTerms` and `msarc.filterTerms` return the modified `msarc` object.

`msarc.saveTerms` returns nothing.

`msarc.getTerms` returns a 3-column data frame: GO ID, GO category name, count of the number of proteins that match this category.

Author(s)

Gord Brown

Examples

```
data(sample_goterms,package="msarc")
## Not run: msarc.saveTerms(sample_goterms)
## Not run: sample_filtered <- msarc.loadTerms(sample_goterms,"terms.txt")
tbl <- msarc.getTerms(sample_goterms)
tbl <- tbl[c("GO:0017076","GO:0030554","GO:0032553","GO:0032555","GO:0032559",
            "GO:0035639","GO:0036094","GO:0097159","GO:0017111","GO:0016462"),]
sample_filtered <- msarc.filterTerms(sample_goterms,tbl)
```

---

msarc.plotCircle *Draw* msarc *Diagram without GO categories*

Description

Given an `msarc` object, draw the corresponding SVG-formatted circle diagram, without GO categories.

Usage

```
msarc.plotCircle(msarc, file = "nocat.svg", col = c(0,255,0))
```

Arguments

`msarc`	an `msarc` object to draw.
`file`	a filename to write the diagram to. Will overwrite existing files without warning.
`col`	a 3-element list of RGB values (0-255).

Details

The main purpose of this function is simply to draw the diagram, without GO categories.

Value

Returns the `msarc` object.

Author(s)

Gord Brown

Examples

```
data('sample_complete',package="msarc")
## Not run: msarc.plotCircle(sample_complete,file="thing.svg")
```

`msarc.plotHeatmap` *Correlation Heat Maps for Mass Spec Experiments*

Description

Given 3 or more mass spec experiments as `msarc` objects, this function draws a heat map of the similarity among the experiments. Similarity may be calculated based on similarity of scores, or simply by the presence or absence of individual UniProt accession id's.

Usage

```
msarc.plotHeatmap(msalist, method = "euclidean", useScore = T, ...)
```

Arguments

`msalist`	a list of `msarc` objects.
`method`	Correlation method to pass to `dist`. Acceptable values are whatever `dist` supports, currently "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". See the documentation for `stats:dist` for details.
`useScore`	If true (the default), use the score of each peptide in computing the correlation. If false, just use the presence or absence of each protein in each experiment.
`...`	Other parameters to pass to `heatmap.2` from package `gplots`.

Details

The samples must have distinct filenames, or this function will fail. You can load the samples from mass spec files, or just explicitly set object$filename for each `msarc` object. These filenames will also be used as labels in the heat map.

Value

the distance matrix calculated by `dist`.

Author(s)

Gord Brown

References

Gregory R. Warnes. Includes R source code and/or documentation contributed by: Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz and Bill Venables (2011). gplots: Various R programming tools for plotting data. R package version 2.10.1. http://CRAN.R-project.org/package=gplots

See Also

See also `heatmap.2` from package `gplots` and `dist` from package `stats`.

Examples

```
data('sample_cluster',package="msarc")
## Not run: msarc.plotHeatmap(sample_cluster)
```

`msarc.plotSVG` *Draw* `msarc` *Diagram*

Description

Given an `msarc` object, draw the corresponding SVG-formatted diagram.

Usage

```
msarc.plotSVG(msarc, file = "msarc.svg")
```

Arguments

`msarc`	an `msarc` object to draw.
`file`	a filename to write the diagram to. Will overwrite existing files without warning.

Details

The main purpose of this function is simply to draw the diagram, after a presumably interesting set of GO terms has been chosen.

Because GO category names get longer as the categories get smaller, there usually isn't room to draw them in a non-overlapping way on the diagram. For sub-categories, single letters are added as category labels, and the names are printed to the R console. The user can add names of their choice, and find a way to make them fit manually.

The object is returned with colors added, in case the user wants to draw multiple diagrams with the same colors for the same categories. Draw the first object, saving it to a variable, then create the second and use `msarc.copyColors` to copy the color palette from the first to the second. Save the second, copy its color palette to the third, and so on. In this way, GO categories will have the same color in all the diagrams. It's a bit tedious, but there isn't an obvious other way to achieve this.

Value

Returns the `msarc` object, augmented with the set of colors used.

Author(s)

Gord Brown

Examples

```
data('sample_complete',package="msarc")
## Not run: msarc.plotSVG(sample_complete,file="thing.svg")
```

---

msarc.subtract *Subtract Control From Experiment*

Description

Removes Uniprot IDs from an `msarc` object if they also occur in a "control" object (or a simple vector of Uniprot IDs).

Usage

```
msarc.subtract(source, control)
```

Arguments

`source`	the `msarc` object of interest
`control`	the background control, either another `msarc` object, or a simple vector of Uniprot IDs

Details

Typically this will be used to remove "background" proteins, such as those obtained from an IgG control, from an experiment, to reveal those proteins which are specific to the experiment of interest. Customarily, a mass spec experiment will include a treated sample of some sort, and a corresponding untreated sample. The peptides that are identified only in the treated sample are taken to be the interesting ones. This function allows the straightforward removal from the treatment object of proteins that also occur in the untreated control.

Value

the modified source `msarc` object.

Author(s)

Gord Brown

Examples

```
# Load a data set and control
data('sample_initial',package="msarc")
data('control_initial',package="msarc")
sample_subtracted <- msarc.subtract(sample_initial,control_initial)
```

---

msarc.tagCloud *Draw Tag Cloud*

Description

Given an msarc object, draw the corresponding tag cloud.

Usage

```
msarc.tagCloud(msarc, pal=NA,...)
```

Arguments

msarc	an msarc object to draw.
pal	an RColorBrewer palette to use for word colours. Default is brewer.pal(10,"Spectral").
...	additional arguments passed to wordcloud.

Details

The main purpose of this function is to draw a tag cloud (word cloud) of the identified proteins, with size indicating Mascot score.

The tag cloud will be drawn on the current device; it is the user's responsibility to open a connection to a file, or other device as applicable.

Value

Does not return a value.

Author(s)

Gord Brown

Examples

```
data('sample_complete',package="msarc")
## Not run: msarc.tagCloud(sample_complete)
```

print.msarc *Display an* msarc *Object*

Description

These functions display an msarc object in a readable form.

Usage

```
## S3 method for class 'msarc'
print(x, ...)
## S3 method for class 'msarc'
summary(object, ...)
```

Arguments

x	an msarc object
object	an msarc object
...	other parameters to print or summary

Details

Invoked automatically by print or summary when applied to an msarc object. summary provides more details.

Value

print.msarc invisibly returns a string representation of the object. summary.msarc returns nothing.

Author(s)

Gord Brown

See Also

See also print and summary.

Examples

```
data('sample_complete',package="msarc")
print(sample_complete)
summary(sample_complete)
```

sample_df *Sample Data for the msarc Package*

Description

These two mass spec data sets are the experiment and control results from a mass spec of a GREB1 ChIP (sample_df) and an IgG ChIP (control_df).

Usage

```
data(sample_df)
data(control_df)
```

Format

Data frames with the following 3 columns:

`uniprot` the UniProt ID of the protein

`id` the symbolic name of the protein

`score` the MASCOT score of the protein

Source

From Mohammed, H., D'Santos, C., Serandour, A. A., Ali, H. R., Brown, G. D., Atkins, A., et al. (2013). Endogenous Purification Reveals GREB1 as a Key Estrogen Receptor Regulatory Factor. Cell reports, 3(2), 342-349. doi:10.1016/j.celrep.2013.01.010

Examples

```
data(sample_df,package="msarc")
data(control_df,package="msarc")
```

sample_initial *Sample Objects for the msarc Package*

Description

These `msarc` objects are used as examples in the `msarc` package.

Usage

```
data(sample_initial)
```

Format

These are `msarc` objects.

Source

From Mohammed, H., D'Santos, C., Serandour, A. A., Ali, H. R., Brown, G. D., Atkins, A., et al. (2013). Endogenous Purification Reveals GREB1 as a Key Estrogen Receptor Regulatory Factor. Cell reports, 3(2), 342-349. doi:10.1016/j.celrep.2013.01.010

Examples

```
data(sample_initial,package="msarc")
data(control_initial,package="msarc")
```

Index

∗Topic **IO**
msarc, 3
msarc.loadMS, 6
msarc.loadTerms, 7
print.msarc, 13
∗Topic **color**
msarc.copyColors, 4
∗Topic **datasets**
sample_df, 14
sample_initial, 14
∗Topic **dplot**
msarc.copyColors, 4
∗Topic **file**
msarc, 3
msarc.loadMS, 6
msarc.loadTerms, 7
∗Topic **hplot**
msarc.plotCircle, 8
msarc.plotHeatmap, 9
msarc.plotSVG, 10
msarc.tagCloud, 12
∗Topic **manip**
msarc.findGOterms, 5
msarc.subtract, 11
∗Topic **methods**
print.msarc, 13
∗Topic **print**
print.msarc, 13

control_df *(*sample_df*)*, 14
control_initial *(*sample_initial*)*, 14

msarc, 3
msarc-package, 2
msarc.copyColors, 4
msarc.filterTerms *(*msarc.loadTerms*)*, 7
msarc.findGOterms, 5
msarc.getTerms *(*msarc.loadTerms*)*, 7
msarc.loadMS, 6
msarc.loadTerms, 7
msarc.plotCircle, 8
msarc.plotHeatmap, 9
msarc.plotSVG, 10
msarc.saveTerms *(*msarc.loadTerms*)*, 7
msarc.subtract, 11
msarc.tagCloud, 12

print, *13*
print.msarc, 13

sample_cluster *(*sample_initial*)*, 14
sample_complete *(*sample_initial*)*, 14
sample_df, 14
sample_goterms *(*sample_initial*)*, 14
sample_initial, 14
sample_subtracted *(*sample_initial*)*, 14
summary, *13*
summary.msarc *(*print.msarc*)*, 13