Package 'PAPi'

April 15, 2020

Type Package

Title Predict metabolic pathway activity based on metabolomics data

Version 1.26.0

Date 2013-03-27

Author Raphael Aggio

Maintainer Raphael Aggio <raphael.aggio@gmail.com>

Description The Pathway Activity Profiling - PAPi - is an R package for predicting the activity of metabolic pathways based solely on a metabolomics data set containing a list of metabolites identified and their respective abundances in different biological samples. PAPi generates hypothesis that improves the final biological interpretation. See Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

License GPL(>= 2)

Depends R (>= 2.15.2), svDialogs, KEGGREST

biocViews ImmunoOncology, MassSpectrometry, Metabolomics

git_url https://git.bioconductor.org/packages/PAPi

git_branch RELEASE_3_10

git_last_commit 43a0675

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

R topics documented:

PAPi-package *Pathway Activity Profiling*

Description

PAPi is an R package that implements the PAPi algorithm published in Aggio et al. 2010. It relates metabolite abundances to metabolic pathway activity. For this, PAPi uses a typical metabolomics data set and the KEGG (Kyoto Encyclopedia of Genes and Genomes) database (http://www.genome.jp/kegg/).

Details

Package:	PAPi
Type:	Package
Version:	0.99.3
Date:	2013-03-27
License:	GPL(>= 2)

PAPi has five functions: buildDatabase, addKeggCodes, papi, papiHtest and papiLine. The main function of PAPi is papi, which uses a metabolomics data set for predicting the activity of metabolic pathways. For this, papi requires the name of each compound to be substituted by its respective KEGG code. KEGG code is a unique compound identifier in KEGG database. The function addKeggCodes then automatically substitutes compounds names by their respective KEGG codes while papi generates a list of metabolic pathways with their predicted metabolic activity. papiHtest performs ANOVA or t-test on results generated by papi. Finally, papiLine generates a line graph of results produced by papi or papiHtest.

Author(s)

Raphael Aggio Maintainer: Raphael Aggio (raphael.aggio@gmail.com)

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

Examples

```
### Building input data ####
Names <- c("Replicates", "C00197", "C05345", "C00031", "C00118", "C00111")
Sample1 <- c("cond1", 0.2, 0.3, 0.8, 1.1, 1.2)
Sample2 <- c("cond1", 0.3, 0.2, 0.6, 1.5, 1.5)
Sample3 <- c("cond1", 0.5, 0.4, 0.7, 1.2, 1.3)
Sample4 <- c("cond2", 1.1, 0.6, 1.2, NA, 0.2)
Sample5 <- c("cond2", 1.0, 0.7, 1.1, NA, 0.3)
Sample6 <- c("cond2", 0.9, 0.7, 1.5, NA, 0.2)

papiData <- data.frame(cbind(Names, Sample1, Sample2, Sample3, Sample4, Sample5,
Sample6), stringsAsFactors = FALSE)

### Applying papi ####
#papiResults <- papi(papiData, save = FALSE)
```

addKeggCodes *In a data frame, it substitutes the name of compounds by their respective KEGG codes.*

Description

addKeggCodes makes use of a user library to automatically substitute compounds names by KEGG compounds codes. KEGG codes are required to use the function papi.

Usage

```
addKeggCodes(inputData, keggCodes, folder, save = TRUE,
output = "data_with_kegg_codes", addCodes = TRUE)
```

Arguments

inputData	when inputData is missing, a dialog box will pop up allowing the user to click-and-point to the comma separated value (CSV) file from which the input data is to be read. Alternatively, inputData can take a character string naming the path to the CSV file to be read or the name of a variable (data frame) containing the metabolomics data (See data(metabolomicsData)).
keggCodes	when keggCodes is missing, a dialog box will pop up allowing the user to click-and-point to the comma separated value (CSV) file containing the KEGG codes library (See Details).
folder	when save = TRUE and folder is missing, a pop up dialog box will be presented to the user. The user can then select the directory to which the results will be saved. Alternatively, folder can take a character string naming the path to the folder where the results must be saved.
save	A logical vector (TRUE or FALSE) defining if the results must be saved into a CSV file.
output	A character string indicating the name of the file containing the results.
addCodes	A logical vector (TRUE or FALSE) defining if missing KEGG codes should be added to the library in use (See Details).

Details

papi is an algorithm to relate metabolite abundances to the activity of metabolic pathways. The input data for papi is a typical metabolomics data set, which is generally organized as a list of metabolites in the first column with their respective abundances in different samples in the following columns (See data(metabolomicsData)). For papi, the name of metabolites MUST be substituted by their respective KEGG codes. The KEGG code of a compound is a unique identifier used by KEGG database. addKeggCodes automatically substitutes the name of compounds by their KEGG codes found in a KEGG code library built and defined by the user. The inputData for addKeggCodes is a data frame containing the name of compounds in the first column and their respective abundances in the different samples in the following columns. See data(metabolomicsData) for an example of inputData. The keggCodes library is a data frame that MUST contain Kegg codes in the first column and their respective compounds names in the second column. See data(keggLibrary) for an example of Kegg codes library. Ideally, the keggCodes library must contain all the compounds potentially identifiable by the analytical technique and protocol in use. When addCodes = TRUE, compounds that are present in the inputData but not present in the KeggCodes library will be reported to the

user. Then, the missing KEGG codes for these compounds can be automatically searched in the KEGG database or they can be manually added to the KeggCodes library. Compounds not present in KEGG database must receive the value 'absent' as KEGG code. Compounds not related to any KEGG code will NOT be analyzed by papi.

Value

addKeggCodes returns a data frame in the same format of the inputData, however, containing KEGG codes instead of the name of compounds in the first column.

Note

Raphael Aggio (raphael.aggio@gmail.com)

Author(s)

Raphael Aggio

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

See Also

buildDatabase, papi, papiHtest and papiLine.

Examples

```
## Building the input data ####
Names <- c("Replicates", "Glucose", "Fructose 6-phosphate",
"Glyceraldehyde 3-phosphate", "Glycerone phosphate", "3-Phospho-D-glycerate")
Sample1 <- c("cond1", 0.8, 0.3, 1.1, 1.2, 0.2)
Sample2 <- c("cond1", 0.6, 0.2, 1.5, 1.5, 0.3)
Sample3 <- c("cond1", 0.7, 0.4, 1.2, 1.3, 0.5)
Sample4 <- c("cond2", 1.2, 0.6, NA, 0.2, 1.1)
Sample5 <- c("cond2", 1.1, 0.7, NA, 0.3, 1.0)
Sample6 <- c("cond2", 1.5, 0.7, NA, 0.2, 0.9)

metabolomicsData <- data.frame(cbind(Names, Sample1, Sample2, Sample3, Sample4,
 Sample5, Sample6), stringsAsFactors = FALSE)

## Building the keggCodes library ####
kegg <- c("C00031", "C05345", "C00118", "C00111", "C00197", "absent")
Name <- c("Glucose", "Fructose 6-phosphate", "Glyceraldehyde 3-phosphate",
"Glycerone phosphate", "3-Phospho-D-glycerate", "Citrate")

keggLibrary <- data.frame(cbind(kegg, Name), stringsAsFactors = FALSE)

### Applying addKeggCodes ####
papiData <- addKeggCodes(metabolomicsData, keggLibrary, save = FALSE, addCodes = FALSE)
```

buildDatabase — *Build a local database containing the required information from KEGG (Kyoto Encyclopedia of Genes and Genomes) database.*

Description

buildDatabase uses the internet connection to access KEGG database and locally install the required data to use papi offline.

Usage

```
buildDatabase(save = TRUE, folder, saveAs)
```

Arguments

save	when save = TRUE, the new local database is saved in the folder defined by folder in addition to the folder R.home("library/PAPi/databases/").
folder	when save = TRUE and folder is missing, a pop up dialog box will be presented to the user. The user can then select the directory to which the results will be saved. Alternatively, folder can take a character string naming the path to the folder where the new database must be saved.
saveAs	A character string defining the name of the new local database.

Details

papi is an algorithm to relate metabolite abundances to the activity of metabolic pathways. For this, PAPi uses the information available in KEGG database. However, applying PAPi online may be considerably time consuming depending on the internet connection available. In addition, KEGG database is constantly updated, which means that results produced by PAPi today may be different from results produced one month earlier. buildDatabase allows users to build local databases containing the required data from KEGG database. It allows to reproduce results at anytime and considerably faster analysis.

Value

buildDatabase stores the local database in R.home("library/PAPi/databases/").

Note

Raphael Aggio (raphael.aggio@gmail.com)

Author(s)

Raphael Aggio

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

See Also

addKeggCodes, papi, papiHtest and papiLine.

Examples

```
#buildDatabase(save = FALSE, saveAs = "MyNewDatabase")
```

---

keggLibrary *An example of the library mapping name of compounds to their respective KEGG codes.*

Description

keggLibrary is an example of the keggCodes library used by the function addKeggCodes, which automates the substitution of compounds names by KEGG codes. keggLibrary is a data frame containing a list of KEGG compounds codes in the first column and their respective names in the second column.

Usage

```
data(keggLibrary)
```

Format

A data frame with 2 columns and n rows, where n is the number of compounds present in the library.

`kegg` a list of KEGG codes for the compounds described in the second column.

`Name` a list of compound names related to the KEGG codes in column one.

Details

For a high-throughput platform, the keggLibrary must contain all the compounds potentially identifiable by the protocol in use. Every time a new compound is added, for example in the GC-MS library, the same compound must be added to the keggLibrary.

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

Examples

```
data(keggLibrary)
print(keggLibrary)
```

metabolomicsData *An example of metabolomics data set used by* addKeggCodes.

Description

metabolomicsData is an example of a typical metabolomics data set, where it has a list of identified metabolites in the first column and their respective abundances in the different samples in the following columns. The first row of metabolomicsData, Replicates, defines to which experimental condition each sample belongs.

Usage

```
data(metabolomicsData)
```

Format

A data frame containing the names of identified metabolites in the first column and their respective abundances in the different samples in the following columns.

`Names` list of identified metabolites

`Sample1` the abundance of each metabolite in Sample 1

`Sample2` the abundance of each metabolite in Sample 2

`Sample3` the abundance of each metabolite in Sample 3

`Sample4` the abundance of each metabolite in Sample 4

`Sample5` the abundance of each metabolite in Sample 5

`Sample6` the abundance of each metabolite in Sample 6

Examples

```
data(metabolomicsData)
print(metabolomicsData)
```

papi *papi - Pathway Activity Profiling*

Description

papi was developed to apply the PAPi method to a metabolomics data set (See references). PAPi relates metabolite abundances to metabolic pathways activities/fluxes in order to generate potential hypothesis for achieving the biological interpretation.

Usage

```
papi(inputData, save = TRUE, folder, output = "papi_results", offline = TRUE, localDatabase = "defau
```

Arguments

inputData	when inputData is missing, a dialog box will pop up allowing the user to click-and-point to the comma separated value (CSV) file from which the data is to be read. Alternatively, inputData can take a character string naming the path to the CSV file to be read or the name of a variable (data frame) containing the metabolomics data (See Details).
save	A logical vector (TRUE or FALSE) defining if the results must be saved into a CSV file.
folder	when save = TRUE and folder is missing, a pop up dialog box will be presented to the user. The user can then select the directory to which the results will be saved. Alternatively, folder can take a character string naming the path to the folder where the results must be saved.
output	A character string indicating the name of the file containing the results.
offline	A logical vector (TRUE or FALSE) defining if the papi analysis should be done offline using a local database (see buildDatabase).
localDatabase	localDatabase may receive the value "default", "choose" or the name of a specific local database. When localDatabase = "default", the default local database is used. When localDatabase = "choose", a list with the available databases is presented to the user. Alternatively, localDatabase can receive the name of the database to be used.

Details

papi is an algorithm to relate metabolite abundances to the activity of metabolic pathways. The inputData for papi is a typical metabolomics data set (see data(metabolomicsData)), which is generally organized as a list of metabolites in the first column with their respective abundances in different samples in the following columns. For papi, the name of metabolites MUST be substituted by their respective KEGG codes. The KEGG code of a compound is a unique identifier used by KEGG database. See data(papiData) for an example of the inputData. The KEGG codes can be found directly at KEGG website (http://www.genome.jp/kegg/) or by using the function addKeggCodes. The first row of the inputData MAY be used to define to which experimental conditions each sample belongs. For this, the first column of the first row must receive the word "Replicates" and the following columns must receive a character string indicating the experimental conditions to which each sample belongs. papi can be applied by simply inserting papi() at the R console. A pop up window will let the user point to a CSV file containing the input data, while another pop up window will let the user indicate the folder where the results are to be saved. Alternatively, the user may use a character string to indicate the path to the CSV file containing the inputData or the user may indicate a R data frame containing the inputData. The argument offline is used to define if papi will use the KEGG database through an internet connection or if papi will use a local database. PAPi packages contains a default local database, which was created on 25/03/2013. The function buildDatabase can be used to create new local databases. The default behavior of papi is to use a local database, as it is considerably faster than using the internet.

Value

papi returns a data frame containing the identified metabolic pathways in the first column and their respective Activity Score for each sample in the following columns (See data(papiResults)).

Note

Raphael Aggio (raphael.aggio@gmail.com)

Author(s)

Raphael Aggio

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

See Also

papiLine, papiHtest and addKeggCodes.

Examples

```
### Building input data ####
Names <- c("Replicates", "C00197", "C05345", "C00031", "C00118", "C00111")
Sample1 <- c("cond1", 0.2, 0.3, 0.8, 1.1, 1.2)
Sample2 <- c("cond1", 0.3, 0.2, 0.6, 1.5, 1.5)
Sample3 <- c("cond1", 0.5, 0.4, 0.7, 1.2, 1.3)
Sample4 <- c("cond2", 1.1, 0.6, 1.2, NA, 0.2)
Sample5 <- c("cond2", 1.0, 0.7, 1.1, NA, 0.3)
Sample6 <- c("cond2", 0.9, 0.7, 1.5, NA, 0.2)

papiData <- data.frame(cbind(Names, Sample1, Sample2, Sample3, Sample4,
Sample5, Sample6), stringsAsFactors = FALSE)

### Applying papi ####
#papiResults <- papi(papiData, save = FALSE)
```

---

papiData — *An example of the data frame produced by addKeggCodes and used by papi.*

Description

papiData is a data frame containing a list of identified metabolites in the first column, in this case represented by their respective KEGG codes, and their abundances in the different samples in the following columns.

Usage

```
data(papiData)
```

Format

papiData is a data frame containing a list of identified metabolites in the first column, in this case represented by their respective KEGG codes, and their abundances in the different samples in the following columns.

`Name` list of identified metabolites, represented by their respective KEGG codes.

`Sample1` the abundance of each metabolite in Sample 1

`Sample2` the abundance of each metabolite in Sample 2

`Sample3` the abundance of each metabolite in Sample 3

`Sample4` the abundance of each metabolite in Sample 4

`Sample5` the abundance of each metabolite in Sample 5

`Sample6` the abundance of each metabolite in Sample 6

Examples

```
data(papiData)
print(papiData)
```

`papiHtest` *Apply ANOVA or t-test to results produced by* `papi`.

Description

papiHtest applies ANOVA or t-test to a data frame generated by papi.

Usage

```
papiHtest(inputData, signif.level = 0.05, log.transform = TRUE,
save = TRUE, folder, StatTest, output, adjust.pValue = TRUE,
method = "bonferroni")
```

Arguments

`inputData`	when inputData is missing, a dialog box will pop up allowing the user to click-and-point to the comma separated value (CSV) file from which the data is to be read. Alternatively, inputData can take a character string naming the path to the CSV file to be read or the name of a variable (data frame) containing the data frame resultant from `papi`.
`signif.level`	A numeric value indicating the p-value cut-off. Every metabolic pathway showing a p-value higher than the signif.level will be removed from the data frame produced by papiHtest.
`log.transform`	A logical vector (TRUE or FALSE) defining if the data should be log-transformed before applying the statistical test.
`save`	A logical vector (TRUE or FALSE) defining if the results must be saved into a CSV file.
`folder`	When save = TRUE and folder is missing, a pop up dialog box will be presented to the user. The user can then select the directory to which the results will be saved. Alternatively, folder can take a character string naming the path to the folder where the results must be saved.
`StatTest`	When StatTest is missing, a t-test is performed if the input data has exactly two experimental conditions and an ANOVA is performed otherwise. Alternatively, StatTest may receive the values "ANOVA", "Anova", "anova", "A" or "a" for ANOVA, or "T-TEST", "T-test", "t-test", "t-TEST", "t" and "T" for t-test.
`output`	A character string indicating the name of the file containing the results. When output is missing, the CSV file produced is named according to the statistical test performed, t-test.csv or ANOVA.csv.
`adjust.pValue`	A logical vector indicating if p-values should be adjusted by the `p.adjust`.
`method`	A character string indicating the method used when applying `p.adjust`. See `p.adjust.methods` for possible methods. Default is 'bonferroni'.

Details

The inputData must have the same format as data(papiResults). papiHtest can be applied leaving all of it arguments as default. In this case, pop up windows will let the user choose the required input data file and the folder to save results.

Value

papiHtest generates a data frame as the input data, however, containing an additional column with the calculated p-values. Metabolic pathways showing a p-value higher than the signif.level are removed.

Note

Raphael Aggio (raphael.aggio@gmail.com)

Author(s)

Raphael Aggio

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

See Also

buildDatabase, papi, papiLine and addKeggCodes.

Examples

```
### Building input data ####
Names <- c("Replicates", "Galactose metabolism", "Glycerolipid metabolism",
"Carbon fixation in photosynthetic organisms", "Fatty acid biosynthesis",
"D-Alanine metabolism")
Sample1 <- c("cond1", 200, 77, 99, 42, 29)
Sample2 <- c("cond1", 236, 72, 80, 39, 14)
Sample3 <- c("cond1", 269, 83, 89, 45, 31)
Sample4 <- c("cond2", 68, 12, NA, 6, 0.006)
Sample5 <- c("cond2", 57, 10, NA, 7, 0.1)
Sample6 <- c("cond2", 58, 13, NA, 8, 0.05)

dataInput <- data.frame(cbind(Names, Sample1, Sample2, Sample3, Sample4,
Sample5, Sample6), stringsAsFactors = FALSE)

### Applying papiHtest ####
papiHtestResults <- papiHtest(dataInput, save = FALSE)
```

papiLine *papiLine generates a line graph of results obtained using* papi *or* papiHtest.

Description

papiLine generates a line graph of the average Activity Score (AS) against the predicted metabolic pathways.

Usage

```
papiLine(inputData, relative = TRUE, save=FALSE, folder, output = "papi_graph",
setRef.cond = FALSE, Ref.cond, color = "auto", cex.xlab, cex.ylab = 1,
position.ylab, margins = c(25, 8, 2, 2), yscale, dot.size = 1.5,
legend.position, cex.legend = 1, graphWidth=3000, graphHeight=2000,
graph.bg="transparent", graph.res = 300)
```

Arguments

inputData	when inputData is missing, a dialog box will pop up allowing the user to click-and-point to the comma separated value (CSV) file from which the data is to be read.
relative	a logical vector (TRUE or FALSE) used to define if the input data should be scaled and inverted before ploting (See Details).
save	A logical vector (TRUE or FALSE) defining if the results must be saved into a png file.
folder	when save = TRUE and folder is missing, a pop up dialog box will be presented to the user. The user can then select the directory to which the results will be saved. Alternatively, folder can take a character string naming the path where the results must be saved.
output	A character string indicating the name of the file containing the results.
setRef.cond	A logical vector (TRUE or FALSE) indicating wether a specific experimental condition must be set as reference (See Details).
Ref.cond	A character string indicating the experimental condition to be used as reference.
color	color defines the colors used for the lines corresponding to each experimental condition. It can receive the values "auto", "manual" or the names of the colors to be used (See Details).
cex.xlab	A numeric string defining the size of the font used for the x axis.
cex.ylab	A numeric string defining the size of the font used for the y axis.
position.ylab	A numeric vector of length 2, giving the x and y coordinates of the y axis label position.
margins	A numerical vector in the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot.
yscale	A numeric vector of length 2, giving the y coordinates ranges.
dot.size	A numeric vector defining the size of the dots used to indicate pathways detected only in some experimental conditions.

`legend.position`
: The x and y co-ordinates defining the position of the legend. They can be specified by keywords ("bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center") or in any way which is accepted by xy.coords (See Details).

`cex.legend`
: A numeric vector indicating the size of the legend.

`graphWidth`
: A numeric vector indicating the width of the graph.

`graphHeight`
: A numeric vector indicating the heigh of the graph.

`graph.bg`
: The color to be used for the background of the graph region.

`graph.res`
: A numeric vector indicating the resolution of the figure containing the line graph.

Details

The inputData must have the same format as the results generated by `papi` or `papiHtest` (See data(papiResults)). The first row of the inputData MAY be used to define to which experimental conditions each sample belongs. For this, the first column of the first row must receive the word "Replicates" and the following columns must receive a character string indicating the experimental condition to which each sample belongs. If there is no information about replicates in the inputData, a pop up dialog box will let the user indicate the number of experimental conditions under study and which sample belongs to each condition. The pathway Activity Score (AS) generated by PAPi is inversely proportional to the metabolic pathway activity, which means that the higher the AS the lower is the potential pathway activity. Thus, when relative = TRUE, the y axis is inverted and rescaled to an experimental condition set as reference. The AS of the reference condition is set to 0 and the AS of the remaining conditions are recalculated in relation to the reference condition. When setRef.cond = FALSE, the experimental condition defined in the second column of the inputData will be used as reference. When setRef.cond = TRUE, Ref.cond must receive the name of the experimental condition to be used as reference. If setRef.cond = TRUE and Ref.cond is missing, a pop up dialog box is presented to the user in order to select the experimental condition to be used as reference. color is used to define the color of the lines representing the average AS of each experimental condition. It may receive the values "auto", "manual" or the name of the colors to be used (e.g. c("blue", "green")). If color = "auto", the colors are automatically defined according to colors()[30*c(1:nlevels(reps))], where nlevel(reps) indicates the number of experimental conditions under study. If color = "manual", a pop up dialog box presents to the user a list of colors to be used. legend.position indicates the position of the legend in the graph. It may receive the values "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". Alternatively, it may receive the exact numerical coordinates, such as c(2,0).

Value

papiLine returns a line graph which can be exported to a png file.

Note

Raphael Aggio (rahael.aggio@gmail.com)

Author(s)

Raphael Aggio

References

Aggio, R.B.M; Ruggiero, K. and Villas-Boas, S.G. (2010) - Pathway Activity Profiling (PAPi): from metabolite profile to metabolic pathway activity. Bioinformatics.

See Also

buildDatabase, papi, papiHtest and addKeggCodes.

Examples

```
### Building input data ####
Names <- c("Replicates", "Galactose metabolism", "Glycerolipid metabolism",
"Carbon fixation in photosynthetic organisms", "Fatty acid biosynthesis",
"D-Alanine metabolism")
Sample1 <- c("cond1", 200, 77, 99, 42, 29)
Sample2 <- c("cond1", 236, 72, 80, 39, 14)
Sample3 <- c("cond1", 269, 83, 89, 45, 31)
Sample4 <- c("cond2", 68, 12, NA, 6, 0.006)
Sample5 <- c("cond2", 57, 10, NA, 7, 0.1)
Sample6 <- c("cond2", 58, 13, NA, 8, 0.05)

dataInput <- data.frame(cbind(Names, Sample1, Sample2, Sample3, Sample4,
Sample5, Sample6), stringsAsFactors = FALSE)

# papiLine(dataInput, relative = TRUE, setRef.cond = TRUE, Ref.cond = "cond1",
# save = FALSE, position.ylab = c(2.2, 0.5))
```

---

papiResults *An example of the data frame produced by papi.*

Description

An example of the data frame produced by papi, where the first column contains the name of the metabolic pathways identified and the following columns contains their respective Activity Score (AS) in the different samples analyzed.

Usage

```
data(papiResults)
```

Format

A data frame where the first column contains the name of the metabolic pathways identified and the following columns contains their respective Activity Score (AS) in the different samples analyzed.

`pathwayname` a list of metabolic pathways identified.

`Sample1` the AS of each metabolic pathway in Sample 1

`Sample2` the AS of each metabolic pathway in Sample 2

`Sample3` the AS of each metabolic pathway in Sample 3

`Sample4` the AS of each metabolic pathway in Sample 4

`Sample5` the AS of each metabolic pathway in Sample 5

`Sample6` the AS of each metabolic pathway in Sample 6

Examples

```
data(papiResults)
print(papiResults)
```

Index