Package ‘apmsWAPP’

February 19, 2015

Type Package

Title Pre- and Postprocessing for AP-MS data analysis based on spectral counts

Version 1.0

Date 2013-03-14

Author Martina Fischer

Maintainer Martina Fischer <fischerm@rki.de>

Description apmsWAPP provides a complete workflow for the analysis of AP-MS data (replicate single-bait purifications including negative controls) based on spectral counts.
It comprises pre-processing, scoring and postprocessing of protein interactions.
A final list of interaction candidates is reported: it provides a ranking of the candidates according to their p-values which allow estimating the number of false-positive interactions.

Depends R (>= 3.0.1)

Imports genefilter, Biobase, seqinr, multtest, gtools, edgeR, DESeq, aroma.light

License LGPL-3

LazyLoad yes

SystemRequirements SAINT_v2.3.4

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-22 14:39:54

R topics documented:

apmsWAPP-package *Pre- and Postprocessing for AP-MS data analysis*

Description

The package **apmsWAPP** provides a complete workflow for the analysis of AP-MS data, based on replicate single-bait purifications including negative controls.
It comprises the three main parts of pre-processing, scoring and postprocessing of interaction proteins:
For pre-processing, five different normalization methods and a filtering procedure is provided.
For scoring protein-protein-interactions, either the method of SAINT or a two-stage-poisson model (TSPM) adapted to AP-MS data can be chosen.
For postprocessing, the user can choose between the permutation-based approach of Westfall&Young (applicable to both, SAINT and TSPM) and the adjustment procedure of Benjamini-Hochberg (applicable to TSPM). Postprocessing results in the generation of p-values for each interaction candidate, allowing to control the number of false-positive interactions.

Details

Package:	apmsWAPP
Type:	Package
Version:	1.0
Date:	2013-03-14
License:	LGPL-3

The two main function calls are: `saint_permF` (framework based on SAINT) and `tspm_apms` (framework based on TSPM).
Note: `saint_permF` can only be executed in a linux environment and SAINT must be installed accordingly. `tspm_apms` is applicable in a windows and a linux environment.

Author(s)

Martina Fischer (fischerm@rki.de)

References

Fischer M, Zilkenat S, Gerlach R, Wagner S, Renard BY. Pre- and Post-Processing Workflow for Affinity Purification Mass Spectrometry Data. *Journal of Proteome Research* 2014.

Choi H, Larsen B, Lin Z-Y, et al. SAINT: probabilistic scoring of affinity purification-mass spectrometry data. *Nature Methods* 2011.

Auer PL, Doerge RW. A two-stage Poisson model for testing RNA-Seq data. *Statistical Applications in Genetics and Molecular Biology* 2011.

`int_mat` *Format transformation of spectral counts*

Description

Transformation of a count matrix into an interaction table (format required for SAINT) and vice versa.

Usage

```
int2mat(IntSaint)

mat2int(mat, baittab)
```

Arguments

`IntSaint` a data.frame. The interaction table as required for SAINT (including zero counts).

`mat` matrix of spectral counts, proteins in rows and samples in columns.

`baittab` a data.frame. The baittable as required for SAINT, classifying control and bait samples.

Details

The *interaction table* consists of four columns: IP name, bait or control name, protein name, spectral count (**note**: a protein which was not detected in one of the samples receives a zero count).
`int2mat` transfers the interaction table into a matrix form. `mat2int` transfers a matrix of spectral counts into the interaction table format defined by SAINT.

Value

Either a matrix of spectral counts or a data.frame representing the interaction table is returned.

Author(s)

Martina Fischer

Examples

```
intfile <- system.file("extdata", "inttable.txt", package="apmsWAPP")
interactiontable <- read.table(intfile)
count.mat <- int2mat(interactiontable)
class(count.mat)
dim(count.mat)
```

norm.inttable *Normalization of spectral count data*

Description

Normalization of spectral counts in bait and control samples based on an AP-MS experiment.

Usage

```
norm.inttable(inttab.mat, baittab,
     norm = c("sumtotal", "upperquartile",
              "DESeq", "TMM", "quantile"))
```

Arguments

inttab.mat	matrix of spectral counts, proteins in rows and samples in columns.
baittab	a data.frame. The baittable as required for SAINT, classifying control and bait samples.
norm	method to normalize the data.

Details

The baittable corresponds to a format as required for SAINT, consisting of three columns: IP name, bait or control name, indicator for bait and control experiment (T=bait purification, C=control). Note that the IP names in the baittable must be in agreement with the sample names.

Five different normalization methods, adapted from microarray and RNA-seq analysis to AP-MS data, are available:
In the ‘sumtotal’ normalization counts are divided by the total number of counts in the sample. The ‘upperquartile’ normalization corrects counts by dividing each count by the 75% quantile of its sample counts. The ‘quantile’ method equalizes the distributions of protein counts across all samples. In the ‘DESeq’ approach by Anders and Huber (2010), counts are divided by the the median of the ratio of its count over its geometric mean across all samples. In the ‘TMM’ approach by Robinson and Oshlack (2010), a scaling factor is computed as the weighted mean of log ratios between chosen test and reference samples.

Value

A list containing the following components:

1	normalized spectral count matrix
2	scaling factors (if available)

Author(s)

Martina Fischer

References

Anders S, Huber W. Differential expression analysis for sequence count data. *Genome Biology* 2010.

Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 2010.

Bolstad BM, Irizarry RA, Astrand M, et al. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 2003.

Dillies M-A, Rau A, Aubert J, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics* 2012.

Examples

```
#input data
intfile <- system.file("extdata", "inttable.txt", package="apmsWAPP")
counts <- int2mat(read.table(intfile))
baitfile <- system.file("extdata", "baittab.txt", package="apmsWAPP")
baittab <- read.table(baitfile)
# Normalization:
norm.counts <- norm.inttable(counts, baittab, norm = "upperquartile")
summary(norm.counts[[1]])
norm.counts[[2]]
```

---

saint_permF *Pre- and Postprocessing for AP-MS data analysis using SAINT*

Description

A complete workflow for the identification of true interaction proteins based on AP-MS data, embedding the scoring method SAINT into a pre- and postprocessing framework.

Usage

```
saint_permF(file_baittable, file_inttable, prottable,
   norm = c("none", "sumtotal", "upperquartile", "DESeq",
            "TMM", "quantile"),
   Filter = TRUE,
   filter.method = c("IQR", "overallVar", "noVar"),
   var.cutoff = NA, limit = 0, intern.norm = FALSE,
   saint.options = "2000 10000 0 1 0")
```

Arguments

`file_baittable`	a character string specifying the pathname of the baittable. see *Details*.
`file_inttable`	a character string specifying the pathname of the interaction table. see *Details*.
`prottable`	a character string specifying the pathname of the protein table. see *Details*.

`norm` method to normalize the data. If `norm="none"`, no normalization of the data is performed.

`Filter` logical value, whether filtering of the data is applied (Default `TRUE`).

`filter.method` method to use for filtering, must be one of `"IQR"`, `"overallVar"` or `"noVar"`, only used when `Filter=TRUE`.

`var.cutoff` percentile (between 0 and 1) or `NA`. Cutoff for filtering the data, defined by a quantile or shortest-interval (=`NA`, Default), only used when `Filter=TRUE`.

`limit` minimal number of expected true interaction proteins in the data.

`intern.norm` logical value. If `TRUE`, normalization is repeated on the filtered data (Default `FALSE`).

`saint.options` parameters set for SAINT.

Details

The input files correspond to the input formats used by SAINT: the baittable, prey- and interaction table in the form of tab-delimited files.
The *baittable* consists of three columns: IP name, bait or control name, indicator for bait and control experiment (T=bait purification, C=control).
The *interaction table* consists of four columns: IP name, bait or control name, protein name, spectral count (**note**: a protein which was not detected in one of the samples receives a zero count).
The *protein table* refers to the preyfile, it consists of three columns: protein names, protein length, protein names or associated gene names (if available).
A more detailed description on the generation of these files is given in Choi et.al. *(Current Protocols in Bioinformatics 2012)*.

Pre-processing comprises normalization and filtering of the data:
Here, it can be chosen from five different normalization methods, adapted from microarray and RNA-seq analysis to AP-MS data. For further details see `norm.inttable`.
The filter consists of a biological filter and a statistical variance filter and aims to remove obvious contaminants from further analysis.
If `filter.method="noVar"`, only the biological filter is conducted. Both are conducted, if `filter.method="IQR"`, here the variance is calculated by the inter-quartile-range, or if `filter.method="overallVar"`, here the variance is calculated across all samples.
The `var.cutoff` defines the fraction of proteins with the lowest overall variance, which are considered as contaminants and are removed. `var.cutoff=NA` refers to a cutoff defined by the mean of the shortest intervall containing 50% of the data (default). Alternatively, a quantile can be set as cutoff, e.g. a cutoff of 0.5 filters 50% of the data showing the smallest overall variance or IQR. see also `varFilter`
The parameter `limit` assures, that filtering results in a number of proteins above the number of expected true interaction proteins.

The corresponding parameters in SAINT `[nburn][niter][lowMode][minFold]` `[normalize]` are set as recommended by SAINT. Further details on the parameter setting can be found in Choi et.al.*(Current Protocols in Bioinformatics 2012)*.

Value

The overall result is reported in the file *WY_Result.csv*:
It is based on the original Saint output ‘unique_interactions’, but additionally Westfall&Young ad-

justed p-values are assigned to each interaction candidate. These p-values control the FWER, allowing to estimate the portion of false-positive interactions.

Different .txt and .xls files are generated, enabling the user to follow the different intermediate results:

1. In case of normalization: normalized count data in form of the interaction table (*txt file*), named after the normalization method and the bait protein (e.g. quantile_bait_IntSaint.txt).
2. In case of filtering: the filtered (and normalized) interaction table (*Inttable_filtered.txt*).
3. The Saint output: 'unique_interactions', reporting the interaction candidates with SAINT scores, calculated on normalized data (file name ending *_orig*), and filtered: (file name ending *_orgF*).
4. Permutation data: scores calculated for each permutation data set (permutation matrix as *perm.avgp.Rata*, *perm.maxp.Rdata*).

Note

SAINT is run as part of the workflow. It is important to note that the function `saint_permF` requires a linux environment and was tested on SAINT version 2.3.4.

Author(s)

Martina Fischer

References

Choi H, Larsen B, Lin Z-Y, et al. SAINT: probabilistic scoring of affinity purification-mass spectrometry data. *Nature Methods* 2011.

Choi H, Liu G, Mellacheruvu D, et al. Analyzing Protein-Protein Interactions from Affinity Purification-Mass Spectrometry Data with SAINT. *Current Protocols in Bioinformatics* 2012.

Anders S, Huber W. Differential expression analysis for sequence count data. *Genome Biology* 2010.

Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 2010.

Bolstad BM, Irizarry RA, Astrand M, et al. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 2003.

Westfall PH, Young SS. Resampling-based multiple testing: examples and methods for p-value adjustment. 1993.

Bourgon R, Gentleman R, Huber W. Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences* 2010.

Examples

```
#input dara
baitfile <- system.file("extdata", "baittab.txt", package="apmsWAPP")
intfile <- system.file("extdata", "inttable.txt", package="apmsWAPP")
protfile <- system.file("extdata", "prottable.txt", package="apmsWAPP")
```

```
# To run this example, a linux environment is required and SAINT needs
# to be installed!
# Important: Define a working directory for storage of the resulting
# files
# Pre-processing: quantile normalization and filtering
# Workflow call:
# saint_permF(baitfile,intfile,protfile, norm="quantile", Filter=TRUE,
#       filter.method="overallVar", var.cutoff=0.3, intern.norm=FALSE)
```

tspm_apms *Workflow for AP-MS data analysis using TSPM*

Description

A complete workflow for the analysis of AP-MS data, using a two-stage-poisson model and a pre- and postprocessing framework.

Usage

```
tspm_apms(counts, baittab,
   norm = c("none", "sumtotal", "upperquartile",
            "DESeq", "TMM", "quantile"),
   Filter = TRUE,
   filter.method = c("IQR", "overallVar", "noVar"),
   var.cutoff = NA, limit = 0,
   adj.method = c("BH", "WY"))
```

Arguments

counts	matrix of spectral counts, proteins in rows and samples in columns.
baittab	a character string specifying the pathname of the baittable. see Details.
norm	method to normalize the data. If norm="none", no normalization of the data is performed.
Filter	logical value, whether filtering of the data is applied (Default TRUE).
filter.method	method to use for filtering, must be one of "IQR", "overallVar" or "noVar", only used when Filter=TRUE.
var.cutoff	percentile (between 0 and 1) or NA. Cutoff for filtering the data, defined by a quantile or shortest-interval (=NA, Default), only used when Filter=TRUE.
limit	minimal number of expected true interaction proteins in the data.
adj.method	method to adjust p-values for multiple testing.

Details

The baittable corresponds to a tab/space delimited file as required for SAINT - consisting of three columns: IP name, bait or control name, indicator for bait and control experiment (T=bait purification, C=control).

Pre-processing comprises normalization and filtering of the data:
Here, it can be chosen from five different normalization methods, adapted from microarray and RNA-seq analysis to AP-MS data. For further details see `norm.inttable`.
The filter consists of a biological filter and a statistical variance filter and aims to remove obvious contaminants from further analysis.
If `filter.method="noVar"`, only the biological filter is conducted. Both are conducted, if `filter.method="IQR"`, here the variance is calculated by the inter-quartile-range, or if `filter.method="overallVar"`, here the variance is calculated across all samples.
The `var.cutoff` defines the fraction of proteins with the lowest overall variance, which are considered as contaminants and are removed. `var.cutoff=NA` refers to a cutoff defined by the mean of the shortest intervall containing 50% of the data (default). Alternatively, a quantile can be set as cutoff, e.g. a cutoff of 0.5 filters 50% of the data showing the smallest overall variance or IQR. see also `varFilter`
The parameter `limit` assures, that filtering results in a number of proteins above the number of expected true interaction proteins.

For postprocessing, two different adjustment procedures are provided for multiple testing: the Benjamini-Hochberg procedure (`"BH"`) (p-values are controlled by FDR), and the permutation approach coupled to the Westfall&Young (`"WY"`) algorithm (p-values are controlled by FWER).

Value

A list containing the following components:

`id` name of the interaction protein

`log.fold.change` a vector containing the estimated log fold changes for each protein

`pvalues` a vector containing the raw p-values for each protein, evaluating the interaction

`padj` a vector containing the p-values after adjusting for multiple testing using the method of Benjamini-Hochberg

`LRT` a vector of Likelihood Ratio statistics, scoring the interaction potential of each protein

`dispersion` a vector of yes/no indicating overdispersion for each protein

`adjusted.p` a vector containing the adjusted p-values using the permutation-based approach of Westfall&Young

`counter` a vector containing the number of exceeding permutation scores using the permutation-based approach of Westfall&Young

`matrix1` (filtered) (normalized) matrix of spectral counts

`matrix2` permutation matrix of scores, permutation runs in columns and proteins in rows

Author(s)

Martina Fischer

References

Fischer M, Zilkenat S, Gerlach R, Wagner S, Renard BY. Pre- and Postprocessing for Affinity Purification Mass Spectrometry Data: More Reliable Detection of Interaction Candidates. *Journal of Proteome Research* 2014.

Auer PL, Doerge RW. A two-stage Poisson model for testing RNA-Seq data. *Statistical Applications in Genetics and Molecular Biology* 2011.

Examples

```
# input data
intfile <- system.file("extdata", "inttable.txt", package="apmsWAPP")
counts <- int2mat(read.table(intfile))
baitfile <- system.file("extdata", "baittab.txt", package="apmsWAPP")
# TSPM with quantile normalization and filtering
tspm.quaF <- tspm_apms( counts, baitfile,
                        norm="quantile", Filter=TRUE,
                        filter.method="overallVar",
                        var.cutoff=0.1, adj.method="WY")
# Results:
# for adjustment with BH:
cat("Number of Proteins with p-value <0.05: ",
length(which(tspm.quaF[[1]]$padj < 0.05) ) )
# for adjustment with WY:
cat("Number of Proteins with p-value <0.05: ",
length(which(tspm.quaF[[2]][,2] <0.05)))
```

varFilter *Filtering of AP-MS data*

Description

The filter consists of a biological filter and a statistical variance filter and aims to remove obvious contaminants in AP-MS data.

Usage

```
varFilter(mat, baittab, func = c("IQR", "overallVar", "noVar"),
          var.cutoff = NA, limit = 0)
```

Arguments

`mat`	matrix of spectral counts, proteins in rows and samples in columns.
`baittab`	a data.frame. The baittable as required for SAINT, classifying control and bait samples.
`func`	method to use for filtering, must be one of `"IQR"`, `"overallVar"` or `"noVar"`.
`var.cutoff`	percentile (between 0 and 1) or `NA`. Cutoff for filtering the data, defined by a quantile or shortest-interval (=`NA`, Default).
`limit`	minimal number of expected true interaction proteins in the data.

Details

If filter.method="noVar", only the biological filter is conducted. The biological and statistical filter are applied, if filter.method="IQR", here the variance is calculated by the inter-quartile-range, or if filter.method="overallVar", here the variance is calculated across all samples.

The var.cutoff defines the fraction of proteins with the lowest overall variance, which are considered as contaminants and are removed. var.cutoff=NA refers to a cutoff defined by the mean of the shortest intervall containing 50% of the data (default). Alternatively, a quantile can be set as cutoff, e.g. a cutoff of 0.5 filters 50% of the data showing the smallest overall variance or IQR.

The parameter limit assures, that filtering results in a number of proteins above the number of expected true interaction proteins.
It is recommended to set the parameters var.cutoff and limit according to biological knowledge, if available.

Value

filtered matrix of spectral counts

Author(s)

Martina Fischer

References

Bourgon R, Gentleman R, Huber W. Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences* 2010.

See Also

shorth

Examples

```
#input data
intfile <- system.file("extdata", "inttable.txt", package="apmsWAPP")
counts <- int2mat(read.table(intfile))
baitfile <- system.file("extdata", "baittab.txt", package="apmsWAPP")
baittab <- read.table(baitfile)
dim(counts)
# Filtering:
counts.filtered <- varFilter(counts, baittab, func = "overallVar",
                             var.cutoff = 0.3, limit = 0)
dim(counts.filtered)
```

Index