Package 'net4pg'

October 13, 2022

Title Handle Ambiguity of Protein Identifications from Shotgun
Proteomics

Version 0.1.1

Maintainer Laura Fancello <laura.fancello@gmail.com>

Description In shotgun proteomics, shared peptides (i.e., peptides that might
originate from different proteins sharing homology, from different
proteoforms due to alternative mRNA splicing, post-translational
modifications, proteolytic cleavages, and/or allelic variants) represent a
major source of ambiguity in protein identifications. The 'net4pg' package
allows to assess and handle ambiguity of protein identifications. It
implements methods for two main applications. First, it allows to represent
and quantify ambiguity of protein identifications by means of graph
connected components (CCs). In graph theory, CCs are defined as the largest
subgraphs in which any two vertices are connected to each other by a path
and not connected to any other of the vertices in the supergraph. Here,
proteins sharing one or more peptides are thus gathered in the same CC
(multi-protein CC), while unambiguous protein identifications constitute CCs
with a single protein vertex (single-protein CCs). Therefore, the proportion
of single-protein CCs and the size of multi-protein CCs can be used to
measure the level of ambiguity of protein identifications. The package
implements a strategy to efficiently calculate graph connected
components on large datasets and allows to visually inspect them.
Secondly, the 'net4pg' package allows to exploit the increasing
availability of matched transcriptomic and proteomic datasets to
reduce ambiguity of protein identifications. More precisely, it
implement a transcriptome-based filtering strategy fundamentally
consisting in the removal of those proteins whose corresponding
transcript is not expressed in the sample-matched transcriptome. The
underlying assumption is that, according to the central dogma of
biology, there can be no proteins without the corresponding
transcript. Most importantly, the package allows to visually inspect
the effect of the filtering on protein identifications and quantify
ambiguity before and after filtering by means of graph connected
components. As such, it constitutes a reproducible and transparent
method to exploit transcriptome information to enhance protein

1

identifications. All methods implemented in the 'net4pg' package are fully described in Fancello and Burger (2022) <doi:10.1186/s13059-022-02701-2>.

License GPL-3

URL https://github.com/laurafancello/net4pg

BugReports https://github.com/laurafancello/net4pg/issues

Depends R (>= 3.6.0)

Imports data.table, graph, magrittr, Matrix, methods, utils

Suggests BiocStyle, ggplot2, igraph, knitr, rmarkdown, roxygen2, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Laura Fancello [aut, cre] (<https://orcid.org/0000-0003-4708-4080>), Thomas Burger [aut, ctb] (<https://orcid.org/0000-0003-3539-3564>)

Repository CRAN

Date/Publication 2022-09-07 09:50:02 UTC

R topics documented:

---

cc_composition | *Get peptides and peptide-to-protein mappings for each connected component*

Description

Get peptides and peptide-to-protein mappings for each connected component. For each connected component, first extract its protein members; then, extract all specific and shared peptides mapping on those protein; finally, extract the subset of incidence matrix representing peptide-to-protein mappings.

Usage

```
cc_composition(cc.proteins, incM)
```

Arguments

cc.proteins   a list of vectors (one for each connected component), each enumerating the proteins members of a connected component.

incM   a logical matrix containing the incidence matrix with its column and row names (respectively, protein and peptide identifiers) names and 0 or 1 values indicating whether or not the peptide maps on the corresponding protein.

Value

a list of two elements: i. a list of vectors (one for each connected component) enumerating peptides mapping on protein members of each connected component; ii. a list of matrices or vectors (one for each connected component) representing peptide-to-protein mappings for each connected component; matrices are used if multiple peptides identify protein members of that connected component, vectors if only a single peptide.

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing the proteome incidence matrix
incM_filename <- system.file("extdata"
                              , "incM_example"
                              , package = "net4pg"
                              , mustWork = TRUE)
rownames_filename <- system.file("extdata"
                                  , "peptideIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
colnames_filename <- system.file("extdata"
                                  , "proteinIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                    , colnames_filename = colnames_filename
                    , rownames_filename = rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
# Generate adjacency matrix describing protein-to-protein mappings
adjM <- get_adj_matrix(incM_reduced)
# Generate graph of protein-to-protein connections and calculate its
# connected component
multProteinCC <- get_cc(adjM)
# For each connected component, extract peptides mapping on its protein
# members and the subset of the incidence matrix describing peptide-to-protein
```

```
# mapping
cc.peptides.incM <- cc_composition(cc.proteins = multProteinCC$cc
                                   , incM = incM)
```

---

cc_stats   *Provide statistics on the CCs size*

Description

Provides the CC size distribution, which is the number of CCs including a given number of protein members, and the proportion of single- vs multi-protein CCs.

Usage

```
cc_stats(incM, cc.proteins, reducedIncM)
```

Arguments

incM   a `logical` `matrix` containing the incidence matrix with its column and row names (respectively, protein and peptide identifiers) names and 0 or 1 values indicating whether or not the peptide maps on the corresponding protein.

cc.proteins   a `list` of `vectors` (one for each connected component), each enumerating the proteins members of a connected component.

reducedIncM   a `logical` value indicating if the input matrix is the original complete incidence matrix or a reduced version of it which only contains proteins with at least one shared peptide and all peptides mapping on such proteins (generated by the [reduce_inc_matrix](#) function.

Value

a `list` of four outputs: i. a `vector` containing all those proteins which belong to single-protein connected component (each protein is a single-protein connected component); ii. an `integer` representing the number of single-protein CCs; iii. an `integer` representing the number of multi-protein CCs; iv. a `data.frame` describing size distribution of connected components (number of connected components per number of protein members)

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file("extdata"
                             , "incM_example"
                             , package = "net4pg"
                             , mustWork = TRUE)
```

```
rownames_filename <- system.file("extdata"
                                , "peptideIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
colnames_filename <- system.file("extdata"
                                , "proteinIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                , colnames_filename = colnames_filename
                , rownames_filename = rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
# Generate adjacency matrix describing protein-to-protein mappings
adjM <- get_adj_matrix(incM_reduced)
# Generate graph of protein-to-protein connections and calculate its
# connected components
multProteinCC <- get_cc(adjM)
# CCs size and percentage of single-vs multi-protein CCs
CCstatsOut <- cc_stats(incM = incM_reduced
                       , cc.proteins = multProteinCC$ccs
                       , reducedIncM = TRUE)
```

---

get_adj_matrix   *Generate adjacency matrix*

Description

Generate an adjacency matrix representing protein-to-protein connections, based on shared peptides. It is generated by cross product of the incidence matrix of peptide-to-protein mappings.

Usage

```
get_adj_matrix(incM)
```

Arguments

incM   a `logical` `matrix` containing the incidence matrix with its column and row
                names (respectively, protein and peptide identifiers) names and 0 or 1 values
                indicating whether or not the peptide maps on the corresponding protein.

Value

a `numeric` `matrix` containing the adjacency matrix, with value >0 or 0 indicating whether or not
two proteins are identified by shared peptide(s)

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file( "extdata"
                           , "incM_example"
                           , package = "net4pg"
                           , mustWork = TRUE)
rownames_filename <- system.file( "extdata"
                                , "peptideIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
colnames_filename <- system.file( "extdata"
                                , "proteinIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename=incM_filename
                 , colnames_filename=colnames_filename
                 , rownames_filename=rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
# Generate adjacency matrix describing protein-to-protein mappings
adjM <- get_adj_matrix(incM_reduced)
```

---

get_cc   *Generate graph and calculate its connected components*

Description

Build a graph of protein-to-protein connections from adjacency matrix and calculate its connected components.

Usage

```
get_cc(adjM)
```

Arguments

adjM   a numerical matrix containing the adjacency matrix, with value >0 or 0 indi-
                cating whether or not two proteins are identified by shared peptide(s)

Value

a list of two elements: i. a graph representing protein-to-protein connections encoded by the adjacency matrix; ii. a list of vectors (one for each connected component) enumerating protein members of each connected component.

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file( "extdata"
                            , "incM_example"
                            , package = "net4pg"
                            , mustWork = TRUE)
rownames_filename <- system.file( "extdata"
                                , "peptideIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
colnames_filename <- system.file( "extdata"
                                , "proteinIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename=incM_filename
                 , colnames_filename=colnames_filename
                 , rownames_filename=rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
# Generate adjacency matrix describing protein-to-protein mappings
adjM <- get_adj_matrix(incM_reduced)
# Generate graph of protein-to-protein connections and calculate its
# connected components
multProteinCC <- get_cc(adjM)
```

---

peptide_stats	*Calculate percentage of shared vs specific peptides*

Description

Read in input the incidence matrix of peptide-to-protein mappings generated from valid proteomic identifications

Usage

```
peptide_stats(incM)
```

Arguments

incM   a `logical` `matrix` containing the incidence matrix with its column and row
              names (respectively, protein and peptide identifiers) and 0 or 1 values indicating
              whether or not the peptide maps on the corresponding protein.

Value

a `list` of three elements: i. number of shared peptides; ii. number of specific peptides; iii. percentage of specific peptides

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file( "extdata"
                              , "incM_example"
                              , package = "net4pg"
                              , mustWork = TRUE)
rownames_filename <- system.file( "extdata"
                                  , "peptideIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
colnames_filename <- system.file( "extdata"
                                  , "proteinIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                  , colnames_filename = colnames_filename
                  , rownames_filename = rownames_filename)
# Calculate percentage of shared vs specific peptides
peptideStatsOut <- peptide_stats(incM = incM)
```

---

plot_cc   *Plot peptide-to-protein mapping graph*

Description

Plot a bipartite subgraph representing the connected component to which the user-selected protein belongs. Peptide-to-protein mappings of that protein and of all other proteins belonging to the same CC are represented. The function takes in input a single Ensembl protein identifier (i.e. ENSPXXX for human, ENSMUSPXXX for mouse), it identifies the connected component it belongs to and plots all peptide-to-protein mappings of that connected component.

Usage

```
plot_cc(prot, cc.proteins, cc.subincM, tagProt, tagContam, incM)
```

Arguments

`prot`	a character vector containing a single Ensembl identifier (i.e. ENSPXXXXXXXXXXX for human, ENSMUSPXXXXXXXXXXX for mouse) of the protein of interest.
`cc.proteins`	a list of vectors (one for each connected component) containing protein members of each connected component.
`cc.subincM`	a list of matrices or vectors (one for each connected component) representing the incidence matrix of peptide-to-protein mappings for each connected component; matrices are used if multiple peptides identify protein members of that connected component, vectors if only a single peptide.
`tagProt`	a character vector reporting the prefix of protein identifiers (for non contaminant proteins)
`tagContam`	a character vector reporting the tag which identifies contaminant proteins
`incM`	a logical matrix containing the incidence matrix with its column and row names (respectively, protein and peptide identifiers) names and 0 or 1 values indicating whether or not the peptide maps on the corresponding protein.

Value

a list of four elements: i. CC identifier; ii. protein members of the CC; iii. peptide members of the CC; iv. bipartite subgraph representing peptide-to-protein mapping of that CC (if multi-protein CC)

Author(s)

Laura Fancello

Examples

```
library(igraph)
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file( "extdata"
                            , "incM_example"
                            , package = "net4pg"
                            , mustWork = TRUE)
rownames_filename <- system.file( "extdata"
                                , "peptideIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
colnames_filename <- system.file( "extdata"
                                , "proteinIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                 , colnames_filename = colnames_filename
                 , rownames_filename = rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
# Generate adjacency matrix describing protein-to-protein mappings
adjM <- get_adj_matrix(incM_reduced)
```

```
# Generate graph of protein-to-protein connections and calculate its
# connected components
multProteinCC <- get_cc(adjM)
# For each connected component, extract peptides mapping on its protein
# members and the subset of the incidence matrix describing
# peptide-to-protein mappings
cc.peptides.incM <- cc_composition(cc.proteins = multProteinCC$cc
                                  , incM = incM)
# Plot bipartite graph representing peptide-to-protein mappings for the
# connected component of the protein of interest (in this toy example protein
# "ENSP261"; note that identifiers are not authentic but made up for the
# example)
subgraphCC <- plot_cc(prot="ENSP261"
        , cc.proteins=multProteinCC$ccs
        , cc.subincM=cc.peptides.incM$cc.subincM
        , tagProt = "ENSP"
        , tagContam="Contam"
        , incM=incM)
plot.igraph(subgraphCC$g
      , layout = layout_as_bipartite
      , edge.width = 1
      , edge.arrow.width = 0.3
      , vertex.size = 10
      , edge.arrow.size = 0.5
      , vertex.size2 = 3
      , vertex.label.cex = 0.8
      , asp = 0.35
      , margin = -0.1) +
title(paste0("Protein ENSP261 in CC #", subgraphCC$cc_id), line = -1)
```

---

read_inc_matrix   *Read incidence matrix of proteomic identifications*

Description

Read in input the incidence matrix of peptide-to-protein mappings generated from valid proteomic
identifications.

Usage

```
read_inc_matrix(incM_filename, colnames_filename, rownames_filename)
```

Arguments

incM_filename   the name of the tab-delimited file containing incidence matrix values; the input
                 incidence matrix must contain along the columns protein identifiers and along
                 the rows peptides; each cell must contain a 1 or 0 value indicating whether or
                 not the peptide maps on the corresponding protein.

colnames_filename

> name of the file containing incidence matrix column names, which are protein identifiers. The file must contain one identifier per line. Protein identifiers must be in Ensembl format (i.e., ENSPXXXXXXXXXXXX for human).

rownames_filename

> name of the file containing incidence matrix row names, which are peptide identifiers. The file must contain one identifier per line. Peptide identifiers can be in any format (i.e. numeric identifiers, amino acid sequences, ...)

Value

a `logical` `matrix` containing the incidence matrix with its column and row names (respectively, protein and peptide identifiers) and 0 or 1 values indicating whether or not the peptide maps on the corresponding protein.

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file( "extdata"
                              , "incM_example"
                              , package = "net4pg"
                              , mustWork = TRUE)
rownames_filename <- system.file( "extdata"
                                  , "peptideIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
colnames_filename <- system.file( "extdata"
                                  , "proteinIDs_incM_example"
                                  , package = "net4pg"
                                  , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                 , colnames_filename = colnames_filename
                 , rownames_filename = rownames_filename)
```

---

reduce_inc_matrix   *Reduce size of incidence matrix for downstream analyses*

Description

Reduce the size of the incidence matrix describing peptide-to-protein mappings to ease downstream analyses. The original incidence matrix is reduced to only contain proteins with at least one shared peptide and all peptides mapping on such proteins. This means that only proteins ambiguously identified are retained, which is the most interesting ones when studying ambiguity of protein identifications.

Usage

```
reduce_inc_matrix(incM)
```

Arguments

incM   a `logical` `matrix` containing the incidence matrix with its column and row
                names (respectively, protein and peptide identifiers) and 0 or 1 values indicating
                whether or not the peptide maps on the corresponding protein.

Value

a `logical` `matrix` containing a smaller incidence matrix (with column and row names respectively
reporting protein and peptide identifiers) and 0 or 1 values indicating whether or not the peptide
maps on the corresponding protein. Only proteins with at least one shared peptide and all peptides
mapping on such protein are reported in such reduced incidence matrix.

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing he proteome incidence matrix
incM_filename <- system.file("extdata"
                            , "incM_example"
                            , package = "net4pg"
                            , mustWork = TRUE)
rownames_filename <- system.file("extdata"
                                , "peptideIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
colnames_filename <- system.file("extdata"
                                , "proteinIDs_incM_example"
                                , package = "net4pg"
                                , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                   , colnames_filename = colnames_filename
                   , rownames_filename = rownames_filename)
# Only retain proteins with at least one shared peptide and all peptides
# mapping on such proteins.
incM_reduced <- reduce_inc_matrix(incM)
```

---

transcriptome_filter   *Perform transcriptome-informed post-hoc filtering*

Description

Implement the transcriptome-informed post-hoc filtering strategy. This strategy aims to reduce the ambiguity of protein identifications by exploiting sample-matched transcriptome information, when available. First, it takes in input the set of transcripts expressed in the sample-matched transcriptome (reported using the transcript identifier in Ensembl format, i.e., ENSTXXXX for human) and removes from proteomic identifications: i. all proteins with no expressed transcripts and peptides exclusively mapping on removed proteins ("all"); or ii. only those exclusively identified by shared peptides and peptides exclusively mapping on removed proteins ("sharedOnly"); or iii. only those exclusively identified by shared peptides, whose peptides are shared with at least one protein with expressed transcript, so they are not to be removed ("sharedNoRemove")

Usage

```
transcriptome_filter(
  incM,
  exprTranscriptsFile,
  proteinToTranscriptFile,
  tagContam,
  remove
)
```

Arguments

incM   a `logical` `matrix` containing the incidence matrix with its column and row names (respectively, protein and peptide identifiers) and 0 or 1 values indicating whether or not the peptide maps on the corresponding protein.

exprTranscriptsFile

the name of the file containing the set of transcripts expressed in the sample-matched transcriptome (one per line). Transcript identifiers must be in the Ensembl format (i.e., ENSTXXXXXXXXXXX for human)

proteinToTranscriptFile

the name of a tab-delimited file with protein identifiers in the first column and the corresponding transcript identifiers in the second column. Protein and transcript identifiers must be in the Ensembl format (i.e. ENSPXXXXXXXXXXX and ENSTXXXXXXXXXXX for human)

tagContam   a `character` vector reporting the tag which identifies contaminant protein

remove   `character` vector indicating whether to remove: i. all proteins with no expressed transcripts and peptides exclusively mapping on removed proteins ("all"); ii. only those exclusively identified by shared peptides and peptides exclusively mapping on removed proteins ("sharedOnly"); iii. only those exclusively identified by shared peptides, whose peptides are shared with at least one protein with expressed transcript, so they are not to be removed ("sharedNoRemove")

Value

a `matrix` representing a filtered incidence matrix of peptide-to-protein mapping obtained by transcriptome-informed filtering.

Author(s)

Laura Fancello

Examples

```
# Read the tab-delimited file containing the proteome incidence matrix
incM_filename <- system.file("extdata"
                             , "incM_example"
                             , package = "net4pg"
                             , mustWork = TRUE)
rownames_filename <- system.file("extdata"
                                 , "peptideIDs_incM_example"
                                 , package = "net4pg"
                                 , mustWork = TRUE)
colnames_filename <- system.file("extdata"
                                 , "proteinIDs_incM_example"
                                 , package = "net4pg"
                                 , mustWork = TRUE)
incM <- read_inc_matrix(incM_filename = incM_filename
                  , colnames_filename = colnames_filename
                  , rownames_filename = rownames_filename)
# Perform transcriptome-informed post-hoc filtering
exprTranscriptsFile <- system.file("extdata"
                                   , "expressed_transcripts.txt"
                                   , package = "net4pg"
                                   , mustWork = TRUE)
protein2transcriptFile <- system.file("extdata"
                                      , "protein_to_transcript"
                                      , package = "net4pg"
                                      , mustWork = TRUE)
incM_filtered <- transcriptome_filter(incM
                          , exprTranscriptsFile = exprTranscriptsFile
                          , proteinToTranscriptFile = protein2transcriptFile
                          , tagContam = "Contam"
                          , remove = "all")
```

Index