Package 'RSeed'

October 7, 2016

Type Package

Title Borenstein Analysis

Version 0.1.60

Date 2016-10-07

Author Claus Jonathan Fritzemeier

Maintainer Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

Description An implementation of the analysis about seed components from Borenstein et.al. 2008.

License GNU General Public License

LazyLoad yes

Depends R (>= 2.15.0), methods, sybil, RBGL, graph

Suggests Rgraphviz

Collate RSeed-class.R borenstein.R

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-07 10:53:07

R topics documented:

`findCurrencyMetabolites` *Find currency metabolites*

Description

Find currency metabolites in a metabolic network by the number of reactions using a metabolite.

Usage

```
findCurrencyMetabolites(object, ...)
```

Arguments

`object`	`object` 'modelorg' object
`...`	`cutoff` lower cutoff (default 20).

Details

Identification is performed as follows: From the binary S-matrix is calculated in how many reactions a metabolite participates. If it participates in more than 'cutoff' reactions, it is returned as currency metabolited.

Value

character vector with the metabolite IDs of the currency metabolites.

Author(s)

Claus Jonathan Fritzemeier
<clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Master Thesis
by Claus Jonathan Fritzemeier
at Heinrich-Heine-Universitaet Duesseldorf,
Department of Bioinformatics
in 2012

Borenstein et. al. 2008

See Also

`modelorg` `RSeed`

Examples

```
data(Ec_core)

findCurrencyMetabolites(Ec_core) # cutoff=20
findCurrencyMetabolites(Ec_core, 10)
```

RSeed-class *Class "RSeed"*

Description

Datastructure to perform Borenstein algorithm and save results.

Objects from the Class

Objects can be created by calls of the form `new("RSeed", model, connectedComponentCutOff, currencyMetabolites)`. This is the startup, where the basic parameters are set. They can also be set later.

Slots

note: this class does only have one slot: an environment. to access data, the usere has to use the methods supplied.

`env`: Object of class `"environment"`

Methods

buildGraph `signature(rs = "RSeed")`: Builds a graph from the stoichiometric matrix. In this process is checked weather the graph is connected. Unconnected parts are removed from the graph, if they are smaller than 'connectedComponentCutOff'. If the parts are greater, there will be an error.

combined_nodes `signature(rs = "RSeed")`: Names of the nodes, which combine metabolites.

combined_nodes<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

confidenceLevel `signature(rs = "RSeed")`: Return a confidence level for every seed compound as defined in the original paper. The threshold is 0.2.

connectedComponentCutOff `signature(rs = "RSeed")`: Return the cut off, until which size it is allowed to remove unconnected components from the graph. Default value is 10.

connectedComponentCutOff<- `signature(rs = "RSeed")`: Set this Value.

currencyMetabolites `signature(rs = "RSeed")`: metabolite IDs, which should be treated as currency metabolites and therefor be removed from the network. If this value is NULL, nothing is removed.

currencyMetabolites<- `signature(rs = "RSeed")`: Sets this slot. The replace Value has to be a character vector. The return value from 'findCurrencyMetabolites' is suitable.

getSourceMetabolites `signature(rs = "RSeed")`: Returns a list, with the metabolite IDs of the as essential identified metabolites. Each list entry represents a source compound.

graph_network `signature(rs = "RSeed")`: Graph representation of the network.

graph_network<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

graph_scc `signature(rs = "RSeed")`: Graph representation of the network with the nodes, which belong in one scc, aggregated to one node.

graph_scc<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

initialize `signature(.Object = "RSeed")`: General initialize method. Automatically called from 'new'.

list_sc `signature(rs = "RSeed")`: Return node names of source compounds.

list_sc<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

makeExperiment `signature(rs = "RSeed")`: Performs the whole analysis. The result is stored in the object and in addition to this a reference is returned, too.

model_changes `signature(rs = "RSeed")`: Returns the names of removed metabolites and reactions. These metabolites were either removed because they had no connection to the biggest part in the graph, or because they were given as source metabolites. Reactions are removed if all participating metabolites were removed.

model_changes<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

model_edited `signature(rs = "RSeed")`: The 'modelorg' after removing metabolites and reactions.

model_edited<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

model_original `signature(rs = "RSeed")`: The 'modelorg' from the beginning.

model_original<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

plot `signature(x = "RSeed", y = "missing")`: Plots the graph_scc. Rgraphviz is needed.

scc_sizes `signature(rs = "RSeed")`: Sizes of the SCCs.

scc_sizes<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

scc `signature(rs = "RSeed")`: Returns a named list, which represents the aggregated nodes for every SCC (strongly connected component).

scc<- `signature(rs = "RSeed")`: Setter method for this slot. This should not used by the user, until u really know, what ur doing!

sourceCompounds `signature(rs = "RSeed")`: Aggregates metabolites from the same SCC into a common node.

show `signature(object = "RSeed")`: Prints a short summary about the Object. (Called from generic 'print')

Note

Most slots of this object are automatically set by the algorithm and contain calculated results. So there is low sence in setting them by ur own.

Nodenames are chosen as follows: 'n' + the number of the metabolite. So n4 would mean, that met_id(model)[4] is the represented metabolite.

Author(s)

Claus Jonathan Fritzemeier
<clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Master Thesis
by Claus Jonathan Fritzemeier
at Heinrich-Heine-Universitaet Duesseldorf,
Department of Bioinformatics
in 2012

Borenstein et. al. 2008

See Also

findCurrencyMetabolites, modelorg

Examples

```
data(Ec_core)
ec <- new("RSeed", Ec_core)

# run the experiment
# calls of
# buildGraph(ec)
# sourceCompounds(ec)
# would do the same in two steps.
makeExperiment(ec)

# look which nodes aggregate
scc(ec)

# look at source metabolites
# (every list entry is a source compound)
getSourceMetabolites(ec)

# plot the graph
plot(ec)
```

Index