

Package ‘Corbi’

June 5, 2016

Version 0.3-4

Title Collection of rudimentary bioinformatics tools

Description Provides a bundle of basic and fundamental bioinformatics tools,
such as network querying and alignment.

ByteCompile TRUE

Depends R (>= 3.0.2)

Imports CRF (>= 0.3-10),
Matrix (>= 1.0-14)

License GPL (>= 2)

URL <http://r-forge.r-project.org/projects/corbi/>

RoxygenNote 5.0.1

R topics documented:

Corbi-package	2
best_subnets	2
column	3
extend_subnets	4
get_shortest_distances	4
get_subnets	5
kappa_score	6
net_align	6
net_query	7
nnzero	9
read_net	10
rmultihyper	10
submatrix	11
write_net	12
Index	13

Corbi-package

Corbi - Bioinformatics analysis tools

Description

This package provides a bundle of bioinformatics analysis tools for biomedical research.

Details

This is a collection of bioinformatics analysis tools developed by WuLab at Academy of Mathematics and Systems Science, Chinese Academy of Sciences.

Network comparison:

- [net_query](#) Network querying method based on conditional random fields
- [net_query_batch](#) Batch processing version of [net_query](#)
- [net_align](#) Network alignment method based on conditional random fields

References

Qiang Huang, Ling-Yun Wu, and Xiang-Sun Zhang. An Efficient Network Querying Method Based on Conditional Random Fields. *Bioinformatics*, 27(22):3173-3178, 2011.

Qiang Huang, Ling-Yun Wu, and Xiang-Sun Zhang. Corbi: A new R package for biological network alignment and querying. *BMC Systems Biology*, 7(Suppl 2):S6, 2013.

best_subnets

The best subnetworks that maximize given objective function

Description

Find the best subnetworks that maximize given objective function. If the size of subnetworks \leq exhaust.size, exact exhaustive searching is applied, otherwise, heuristic searching algorithm is used.

Usage

```
best_subnets(func, net.matrix, max.size = 10, exhaust.size = 5,
  max.top = 10000)
```

Arguments

func	The objective function to maximize
net.matrix	The adjacent matrix of network
max.size	The maximal size of subnetworks
exhaust.size	The maximal size of subnetworks that use exhaustive searching strategy
max.top	The maximal number of top candidates kept for evaluation of next size, used in heuristic searching strategy

Value

A list with the following two components:

subnets	The list of top subnetworks in different sizes
obj.values	The list of objective values of corresponding subnetworks

See Also

get_subnets, extend_subnets

Examples

```
library(Corbi)
net <- matrix(FALSE, nrow=10, ncol=10)
net[sample.int(100, 20)] <- TRUE
net <- net | t(net)
func <- function(subnet) max(subnet) - min(subnet)
result <- best_subnets(func, net, 5)
```

column	<i>Extract a column from a matrix</i>
--------	---------------------------------------

Description

Extract a specified column from a sparse matrix rapidly

Usage

```
column(m, i)
```

Arguments

m	The matrix
i	The column index

Details

This function implements faster column extraction algorithm for the [CsparseMatrix](#) class in the package **Matrix**.

Value

This function will return the specified column as a vector of corresponding type.

extend_subnets	<i>All subnetworks by overlapping two subnetworks</i>
----------------	---

Description

Enumerate all possible subnetworks of desired size by overlapping two sets of subnetworks of size s_1 and s_2 . The desired size should be between $\max(s_1, s_2)+1$ and s_1+s_2-1 . Invalid desired size will be replaced by the minimum allowed value $\max(s_1, s_2)+1$.

Usage

```
extend_subnets(subnet1, subnet2, size = 0)
```

Arguments

subnet1	The matrix representing the first set of subnetworks
subnet2	The matrix representing the second set of subnetworks
size	The desired size of extended subnetworks

Value

A matrix represents the extended subnetworks, in which each row represents a subnetwork.

Examples

```
library(Corbi)
net <- matrix(FALSE, nrow=10, ncol=10)
net[sample.int(100, 20)] <- TRUE
net <- net | t(net)
subnets <- get_subnets(net, 3)
subnets[[4]] <- extend_subnets(subnets[[3]], subnets[[2]], 4)
```

get_shortest_distances	<i>Calculate shortest distances of unweighted network</i>
------------------------	---

Description

Calculate all pairs of shortest distances of unweighted network

Usage

```
get_shortest_distances(net.matrix, source.nodes = rep_len(TRUE,
  dim(net.matrix)[1]))
```

Arguments

net.matrix	Logical adjacency matrix of given unweighted network
source.nodes	Logical vector to indicate the source nodes that need to calculate the shortest distances

Details

This function calculates all pairs of shortest distances of unweighted network by using breadth-first-search (BFS) algorithm.

Value

This function will return the shortest distance matrix, where the element $[i, j]$ is the shortest distance between node i and j . Value -1 means unreachable. If `source.nodes[i]` equals `FALSE`, the shortest distance from i to other nodes will not be calculated and the row i will be all -1.

get_subnets	<i>All subnetworks with limited size</i>
-------------	--

Description

Enumerate all subnetworks of size $\leq \text{max.size}$ from given network.

Usage

```
get_subnets(net.matrix, max.size = 2)
```

Arguments

net.matrix	The adjacent matrix of network
max.size	The maximal size of subnetworks

Value

A list of generated subnetworks, with element $\text{\$i\$}$ corresponds the subnetworks of size $\text{\$i\$}$. Each element is a matrix, in which each row represents a subnetwork.

Examples

```
library(Corbi)
net <- matrix(FALSE, nrow=10, ncol=10)
net[sample.int(100, 20)] <- TRUE
net <- net | t(net)
subnets <- get_subnets(net, 3)
```

kappa_score	<i>Cohen's kappa score</i>
-------------	----------------------------

Description

Calculate Cohen's kappa score for two vectors.

Usage

```
kappa_score(x1, x2)
```

Arguments

x1	The first logical vector
x2	The second logical vector

Details

This function calculate Cohen's kappa score for two logical vectors.

Value

The Cohen's kappa score

net_align	<i>Network alignment method based on conditional random fields</i>
-----------	--

Description

Find the maximal matching subnetworks from a target network for a query network based on the conditional random fields (CRF) model.

Usage

```
net_align(query.net, target.net, node.sim, query.type = 4, delta.d = 1e-10,
  delta.c = 0.5, delta.e = 1, delta.s = 1, output = "result.txt")
```

Arguments

query.net	The input file name of the query network.
target.net	The input file name of the target network.
node.sim	The input file name of the node similarity scores between the query network and the target network.
query.type	The querying network type: 1 - general, 2 - chain, 3 - tree, 4 - heuristic.
delta.d	The parameter delta.d is a parameter for deletions.
delta.c	The parameter delta.c is a parameter for consecutive deletions.
delta.e	The parameter delta.e is a parameter for single deletion.
delta.s	The parameter delta.s is a parameter for insertions.
output	The suffix of output file name. The output contains two files in the working directory. One is the matching nodes and edges between query network and target network, the other is the unique matching node pairs.

Details

This is an approach for network alignment problem based on conditional random field (CRF) model which uses the node similarity and structure information equally. This method is based on our network querying method [net_query](#). This method uses an iterative strategy to get the one-to-one map between the query network and target network.

More details can be seen in [net_query](#).

References

Qiang Huang, Ling-Yun Wu, and Xiang-Sun Zhang. CNetA: Network alignment by combining biological and topological features. In Proceedings of 2012 IEEE International Conference on Systems Biology (ISB), 220-225, IEEE, 2012.

Qiang Huang, Ling-Yun Wu, and Xiang-Sun Zhang. Corbi: A new R package for biological network alignment and querying. BMC Systems Biology, 7(Suppl 2):S6, 2013.

Examples

```
## Not run:
library(Corbi)

## An example: "querynet.txt", "targetnet.txt", "nodesim.txt" are
## three input files in the working directory
net_align("querynet.txt", "targetnet.txt", "nodesim.txt")

## End(Not run)
```

net_query

Network querying method based on conditional random fields

Description

Find the best matching subnetworks from a large target network for small query networks based on the conditional random fields (CRF) model.

Usage

```
net_query(query.net, target.net, node.sim, query.type = 4, delta.d = 1e-10,
  delta.c = 0.5, delta.e = 1, delta.s = 1, output = "result.txt")

net_query_batch(query.nets, target.net, node.sim, query.type = 4,
  delta.d = 1e-10, delta.c = 0.5, delta.e = 1, delta.s = 1,
  output = "result.txt")
```

Arguments

query.net	The input file name of the query network.
target.net	The input file name of the target network.
node.sim	The input file name of the node similarity scores between the query network and the target network.

query.type	The querying network type: 1 - general, 2 - chain, 3 - tree, 4 - heuristic.
delta.d	The parameter delta.d is a parameter for deletions.
delta.c	The parameter delta.c is a parameter for consecutive deletions.
delta.e	The parameter delta.e is a parameter for single deletion.
delta.s	The parameter delta.s is a parameter for insertions.
output	The suffix of output file name.
query.nets	The vector of input file names of the query networks.

Details

This is an approach for network querying problem based on conditional random field (CRF) model which can handle both undirected and directed networks, acyclic and cyclic networks, and any number of insertions/deletions.

When querying several networks in the same target network, [net_query_batch](#) will save much time.

- query.net: The query network file is written as follows:
v1 v2 v3 v4 v5
v3 v4
...
where v1, v2, v3, v4, v5 ... are the nodes' names and each line indicates there are edges between the first node and other nodes in the line. For example, the first line denotes 4 edges: (v1, v2), (v1, v3), (v1, v4), and (v1, v5).
- target.net: The format of this file is the same as the query network file.
- node.sim: This similarity file's format is as follows:
v1 V1 s1
v1 V2 s2
...
v1 is the node from the query network, V1 is the node from the target network, s1 is the similarity score between the node v1 and V1, and so on.
- query.type: If query.type = 1, the loopy belief propagation (LBP) algorithm will be applied, which is an approximate algorithm for a general graph with loops. If the query is a chain or tree, there are exact algorithms. Set query.type = 2 when the query is a chain, and query.type = 3 when the query is a tree. The heuristic algorithm will be used when query.type = 4, which will try the exact algorithm (junction tree algorithm) first and resort to LBP algorithm when the exact algorithm failed. The default value is 4.
- delta.d: The smaller delta.d is, the heavier penalty for deletions.
- delta.c: The smaller delta.c is, the heavier penalty for consecutive deletions.
- delta.e: The smaller delta.e is, the heavier penalty for single deletion.
- delta.s: The larger delta.s indicates heavier penalty for insertions.

References

Qiang Huang, Ling-Yun Wu, and Xiang-Sun Zhang. An Efficient Network Querying Method Based on Conditional Random Fields. *Bioinformatics*, 27(22):3173-3178, 2011.

Examples

```
## Not run:
library(Corbi)

## An example: "querynet.txt", "targetnet.txt", "nodesim.txt" are
## three input files in the working directory
net_query("querynet.txt", "targetnet.txt", "nodesim.txt", query.type=3)

## End(Not run)

## Not run:
## Batch example
net_query_batch(c("querynet.txt", "querynet2.txt"),
  "targetnet.txt", "nodesim.txt", query.type=3)

## End(Not run)
```

nnzero

The number of non-zero values of a submatrix

Description

Return the number of non-zero values of the specified submatrix of a given sparse matrix rapidly

Usage

```
nnzero(m, rows = 1:dim(m)[1], cols = 1:dim(m)[2])
```

Arguments

m	The matrix
rows	The integer vector of row index(es) or logical vector indicated the selected rows
cols	The integer vector of column index(es) or logical vector indicated the selected cols

Details

This function implements faster calculation algorithm for the [CsparseMatrix](#) and [RsparseMatrix](#) class in the package **Matrix**.

Value

This function will return the number of non-zero values in the specified submatrix.

read_net	<i>Read network information from text file</i>
----------	--

Description

Read the network information from a text file with specific format.

Usage

```
read_net(file)
```

Arguments

file	The name of text file
------	-----------------------

Details

This function reads the network information from a text file with specific format: each line contains two strings separated by spaces, which correspond to the names of two end points of one edge in the network.

Value

A list with the following components:

size	The number of network nodes
node	The vector of network node names
matrix	The logical adjacency matrix

See Also

[write_net](#)

rmultihyper	<i>The Multivariate Hypergeometric Distribution</i>
-------------	---

Description

Generate random variables for the multivariate hypergeometric distribution

Usage

```
rmultihyper(n, k, m)
```

Arguments

n	The number of observations.
k	The total number of balls drawn from the urn.
m	The integer vector containing the number of balls of each color in the urn. Length of vector is the number of colors.

Details

This function generates random variables for the multivariate hypergeometric distribution by iteratively calling hypergeometric random variable generator [rhyper](#).

Value

This function will return a matrix of `length(m)` rows and `n` columns, and each column contains the number of balls of each color drawn from the urn.

See Also

[rhyper](#)

submatrix	<i>Extract a submatrix from a matrix</i>
-----------	--

Description

Extract a specified submatrix from a sparse matrix rapidly

Usage

```
submatrix(m, rows, cols)
```

Arguments

<code>m</code>	The matrix
<code>rows</code>	The integer vectors of row index(es)
<code>cols</code>	The integer vectors of column index(es)

Details

This function implements faster submatrix extraction algorithm for the [CsparseMatrix](#) class in the package **Matrix**.

Value

This function will return the specified submatrix as a matrix of corresponding type.

write_net	<i>Write network information to text file</i>
-----------	---

Description

Write the network information to a text file with specific format.

Usage

```
write_net(net, file)
```

Arguments

net	A list as returned by read_net
file	The name of text file

Details

This function writes the network information to a text file with specific format: each line contains two strings separated by spaces, which correspond to the names of two end points of one edge in the network.

See Also

[read_net](#)

Index

*Topic **package**

Corbi-package, [2](#)

best_subnets, [2](#)

column, [3](#)

Corbi (Corbi-package), [2](#)

Corbi-package, [2](#)

CsparseMatrix, [3](#), [9](#), [11](#)

extend_subnets, [4](#)

get_shortest_distances, [4](#)

get_subnets, [5](#)

kappa_score, [6](#)

net_align, [2](#), [6](#)

net_query, [2](#), [7](#), [7](#)

net_query_batch, [2](#), [8](#)

net_query_batch (net_query), [7](#)

nnzero, [9](#)

read_net, [10](#), [12](#)

rhyper, [11](#)

rmultihyper, [10](#)

RsparseMatrix, [9](#)

submatrix, [11](#)

write_net, [10](#), [12](#)