

Package ‘Corbi’

October 31, 2011

Version 0.2-1

Date 2011-10-30

Title Corbi - A collection of bioinformatics tools

Author Ling-Yun Wu <wulingyun@gmail.com>

Author@R c(person("Ling-Yun", "Wu", email = "wulingyun@gmail.com"), person("Qiang", "Huang", email = "hq04405114@gmail.com"))

Maintainer Ling-Yun Wu <wulingyun@gmail.com>

Imports CRF (>= 0.2-0)

Depends R (>= 2.12.0)

Description Bioinformatics tools developed by APORC

License GPL (>= 2)

R topics documented:

net.query	1
Index	4

net.query	<i>Network querying method based on conditional random fields</i>
-----------	---

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=1,
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=1,
delta.d=1e-10, delta.c=0.5, delta.e=1, delta.s=1,
output="result.txt")
```

Arguments

<code>query.net</code>	The input file name of the query network.
<code>query.nets</code>	The vector of input file names of the query networks.
<code>target.net</code>	The input file name of the target network.
<code>node.sim</code>	The input file name of the node similarity scores between the query network and the target network.
<code>query.type</code>	The querying network type: 1 - general, 2 - chain, 3 - tree.
<code>delta.d</code>	The parameter <code>delta.d</code> is a parameter for deletions.
<code>delta.c</code>	The parameter <code>delta.c</code> is a parameter for consecutive deletions.
<code>delta.e</code>	The parameter <code>delta.e</code> is a parameter for single deletion.
<code>delta.s</code>	The parameter <code>delta.s</code> is a parameter for insertions.
<code>output</code>	The suffix of output file name.

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 the query is a general graph with loops, we set `query.type = 1`. If the query is a chain, we set `query.type = 2`. If the query is a tree, we set `query.type = 3`. The default value is 1.
- `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*, 2011, doi:10.1093/bioinformatics/BTR524.

Examples

```
library(Corbi)

## An example: "quenet.txt", "tarnet.txt", "queblast.txt" are
## three input files in the working directory
net.query("quenet.txt", "tarnet.txt", "queblast.txt", query.type=3)

## Batch example
net.query.batch(c("quenet.txt", "quenet2.txt", "quenet3.txt"),
               "tarnet.txt", "queblast.txt", query.type=3)
```

Index

`net.query`, [1](#)
`net.query.batch`, [2](#)