

Package ‘QUBIC’

February 16, 2015

Version 1.0

Date 2015-02-01

Author Qin Ma, Yu Shang, Ying Xu

Maintainer

Yu Shang <yushang@uga.edu>, Qin Ma <maqin@uga.edu>, Ying Xu <xyn@bmb.uga.edu>

Title QUBIC is a Qualitative BiCluster Algorithms for Expression Data Analysis

Description This package provides a biclustering program for microarray data. For a set of genes and a set of conditions, the program outputs a block-like structure which shows uniform pattern within the block, the block would contain only subsets of all given genes under subsets of all given conditions.

Depends R (>= 3.0.3), Rcpp (>= 0.11.3)

Type Package

License GPL (>= 2)

LinkingTo Rcpp

LazyLoad yes

LazyData true

R topics documented:

QUBIC-package	1
Ddata	3
Dtoy	3
getBC	4
QUBIC	5
Index	8

QUBIC-package	<i>QUBIC is a Qualitative BiClustering Algorithm for Expression Data Analysis</i>
---------------	---

Description

Biclustering extends the traditional clustering techniques by attempting to find (all) subgroups of genes with similar expression patterns under to-be-identified subsets of experimental conditions when applied to gene expression data. Still the real power of this clustering strategy is yet to be fully realized due to the lack of effective and efficient algorithms for reliably solving the general biclustering problem. We report a QQualitative BIClustering algorithm (QUBIC) that can solve the biclustering problem in a more general form, compared to existing algorithms, through employing a combination of qualitative (or semi-quantitative) measures of gene expression data and a combinatorial optimization technique. One key unique feature of the QUBIC algorithm is that it can identify all statistically significant biclusters including biclusters with the so-called 'scaling patterns', a problem considered to be rather challenging; another key unique feature is that the algorithm solves such general biclustering problems very efficiently, capable of solving biclustering problems with tens of thousands of genes under up to thousands of conditions in a few minutes of the CPU time on a desktop computer. We have demonstrated a considerably improved biclustering performance by our algorithm compared to the existing algorithms on various benchmark sets and data sets of our own.

Certain parts of the code uses open-source data structure library codes, including:

- fib <http://resnet.uoregon.edu/~gurney_j/jmpc/fib.html>, copyright information in fib.c
- Mark A. Weiss's data structure codes <<http://www.cs.fiu.edu/~weiss/>>

The main function of the package is `qubic()`, and the more details you can find in [QUBIC-method](#)

Details

Package:	QUBIC
Type:	Package
Version:	1.0
Date:	2015-02-11
Depends:	R (>= 3.0.3), Rcpp (>= 0.11.3)
LinkingTo:	Rcpp
License:	GPL (>= 2)
LazyLoad:	yes
LazyData:	true

Author(s)

Qin Ma (UGA) <<maqin@uga.edu>>
 Yu Shang (JLU & UGA) <<yushang@uga.edu>>
 Ying Xu (JLU & UGA) <<xyn@bmb.uga.edu>>

Maintainer: Yu Shang (JLU & UGA) <<yushang@uga.edu>>

References

Li, Ma, et al. (2009) *QUBIC: a qualitative biclustering algorithm for analyses of gene expression data* Nucleic Acids Research 07/2009 37(15):e101

<https://code.google.com/p/qubic-biclustering/>

Zhou, Ma, et al. (2012) *QServer: a biclustering server for prediction and assessment of co-expressed gene clusters* PLoS ONE 01/2012 7(3):e32660

See Also

[QUBIC-method](#) [getBC-method](#)

Ddata	<i>A continuous data of Ecoli</i>
-------	-----------------------------------

Description

The data is a matrix to test the continuous data.

Usage

```
data(data)
```

Format

The format is: num [1:1500, 1:466] 9.08 7.12 7.18 7.9 6.8 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:1500] "b4634" "b3241" "b3240" "b3243"\$: chr [1:466] "dinI_U_N0025" "dinP_U_N0025" "lexA_U_N0025" "lon_U_N0025" ...

See Also

[QUBIC-package](#) [QUBIC-method](#)

Examples

```
data(data)
results = qubic(data);
```

Dtoy	<i>A discrete data to test the argument -d</i>
------	--

Description

The toy is a matrix to test the discrete data.

Usage

```
data(toy)
```

Format

The format is: int [1:20, 1:12] 1 1 0 0 0 0 1 0 1 0 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:20] "G1" "G2" "G3" "G4"\$: chr [1:12] "C1" "C2" "C3" "C4" ...

See Also

[QUBIC-package QUBIC-method](#)

Examples

```
data(toy)
results = qubic(toy, d = "T");
```

getBC

Get the details of the bicluster

Description

getBC is a function to get the bicluster of QUBIC.
getBC will return the gene names and condition names of the bicluster you give.

```
thisBC = getBC(results, ...);
```

Arguments

results	The result of QUBIC.
numBC	Which bicluster you want to see.

Details

```
results = qubic(data, file = "rQUBIC", q = 0.06, c = 0.95, f = 1, k = 2, r = 1, o = 100, d = "F");
thisBC = getBC(results, numBC = 1);
```

Value

The R function, getBC returns an object of list:

genes	The gene names in this bicluster.
conds	The condition names in this bicluster.
num_gene	The number of genes in this bicluster.
num_cond	The number of conditions in this bicluster.

Author(s)

Qin Ma (UGA) <<maqin@uga.edu>>
Yu Shang (JLU & UGA) <<yushang@uga.edu>>

Ying Xu (JLU & UGA) <<xyn@bmb.uga.edu>>

Maintainer: Yu Shang (JLU & UGA)<<yushang@uga.edu>>

References

Li, Ma, et al. (2009) *QUBIC: a qualitative biclustering algorithm for analyses of gene expression data* Nucleic Acids Research 07/2009 37(15):e101

<https://code.google.com/p/qubic-biclustering/>

Zhou, Ma, et al. (2012) *QServer: a biclustering server for prediction and assessment of co-expressed gene clusters* PLoS ONE 01/2012 7(3):e32660

See Also

[QUBIC-package](#) [QUBIC-method](#)

Examples

```
data(toy);
# load a discrete data named toy

results = qubic(toy, file = "toy", d = "T");
# run qubic to deal with the discrete data

thisBC = getBC(results, numBC = 3);
# get the gene and condition names of the 3rd bicluster.
```

QUBIC

Qualitative BiClustering Algorithm

Description

QUBIC is a name short for QUalitative BiClustering algorithm, that can solve the biclustering problem in a more general form.

The fuction qubic is the main function of the package. It calculates the bicluster in a data matrix, and it also discrete in a data matrix.

```
results = qubic(data, ...);
```

Arguments

<code>data</code>	Data matrix.
<code>file</code>	-file should be a string. The filename you want the QUBIC to write to. default: "rQUBIC"
<code>q</code>	-q should be (0, 0.5]. Use quantile discretization for continuous data. default: 0.06 (see details in Method section in paper)
<code>c</code>	-c should be (0.5, 1]. Consistency level of the block (0.5-1.0], the minimum ratio between the number of identical valid symbols in a column and the total number of rows in the output. default: 0.95
<code>f</code>	-f should be [0, 1.0]. Filtering overlapping blocks. default: 1 (do not remove any blocks)
<code>k</code>	-k should be ≥ 2 . Minimum column width of the block. default: 5% of columns, minimum 2 columns
<code>r</code>	-r should be a number. The number of ranks as which we treat the up(down)-regulated value when discretization. default: 1
<code>o</code>	-o should be a number. Number of blocks to report. default: 100
<code>d</code>	-d should be either ("True", "False"). Discrete data, where user should send their processed data to different value classes, see above. default: False

Details

`results = qubic(data, file = "rQUBIC", q = 0.06, c = 0.95, f = 1, k = 2, r = 1, o = 100, d = "F");`

Value

The R function, `qubic` returns an object of `list`:

<code>Discrete</code>	The discrete data of your given matrix.
<code>ROWS_COLS</code>	The rows and cols of the discrete data.
<code>BC</code>	The number of genes and conditions of every bicluster.
<code>WriteToFile</code>	The filename you have written to.
<code>ArgumentList</code>	The parameters you have given to.

The R function, `qubic` writes server files to your Working Directory, being the same as the C-code QUBIC does. (see details in paper)

Author(s)

Qin Ma (UGA) <<maqin@uga.edu>>
 Yu Shang (JLU & UGA) <<yushang@uga.edu>>
 Ying Xu (JLU & UGA) <<xyn@bmb.uga.edu>>

Maintainer: Yu Shang (JLU & UGA)<<yushang@uga.edu>>

References

Li, Ma, et al. (2009) *QUBIC: a qualitative biclustering algorithm for analyses of gene expression data* Nucleic Acids Research 07/2009 37(15):e101

<https://code.google.com/p/qubic-biclustering/>

Zhou, Ma, et al. (2012) *QServer: a biclustering server for prediction and assessment of co-expressed gene clusters* PLoS ONE 01/2012 7(3):e32660

See Also

[QUBIC-package getBC-method data toy](#)

Examples

```
data(data);
# load the example matrix

results = qubic(data);
# run QUBIC with default parameters
# results is a variable of list with,
#   the discrete data itself,
#   the rows and cols of discrete data,
#   the filename you want the QUBIC to write the results to,
#   the argument list you sent to the QUBIC,
# the more details of results, you will see in your Working Directory,
# the name of that is what you sent to QUBIC (DEFAULT "rQUBIC"),

thisBC = getBC(results, numBC = 1);
# get the gene and condition names of the biggest, the 1st bicluster.

results = qubic(data, file = "AnotherExample");
# QUBIC will writing to your Working Directory with "AnotherExample" as file name

results = qubic(data, file = "Example", q = 0.06, c = 0.95, f = 1, k = 2, r = 1, o = 100, d = "F");
# run QUBIC with full parameters

data(toy);
# load a discrete data named toy

results = qubic(toy, file = "toy", d = "T");
# run qubic to deal with the discrete data
```

Index

- *Topic **bicluster**
 - getBC, [4](#)
 - QUBIC, [5](#)
 - QUBIC-package, [1](#)
- *Topic **cluster**
 - getBC, [4](#)
 - QUBIC, [5](#)
 - QUBIC-package, [1](#)
- *Topic **continuous**
 - Ddata, [3](#)
- *Topic **datasets**
 - Ddata, [3](#)
 - Dtoy, [3](#)
- *Topic **discrete**
 - Dtoy, [3](#)
- *Topic **matrix**
 - Ddata, [3](#)
 - Dtoy, [3](#)
- data, [7](#)
- data (Ddata), [3](#)
- Ddata, [3](#)
- Dtoy, [3](#)
- getBC, [4](#)
- getBC-method (getBC), [4](#)
- QUBIC, [5](#)
- qubic (QUBIC), [5](#)
- QUBIC-method (QUBIC), [5](#)
- QUBIC-package, [1](#)
- toy, [7](#)
- toy (Dtoy), [3](#)