Bioinformatics, 33(3), 2017, 450–452
doi: 10.1093/bioinformatics/btw635
Advance Access Publication Date: 6 October 2016
Applications Note

Gene expression

QUBIC: a bioconductor package for qualitative biclustering analysis of gene co-expression data

Yu Zhang[1,2,†], Juan Xie[3,4,†], Jinyu Yang[3,4], Anne Fennell[4,5], Chi Zhang[6] and Qin Ma[3,4,5,*]

[1]College of Computer Science and Technology, Jilin University, Changchun, China, [2]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun, China, [3]Department of Mathematics and Statistics, South Dakota State University, Brookings, SD, USA, [4]Department of Agronomy, Horticulture and Plant Science, South Dakota State University, Brookings, SD, USA, [5]BioSNTR, Brookings, SD, USA and [6]Center for Computational Biology and Bioinformatics and Department of Medical and Molecular Genetics, Indiana University School of Medicine, Indianapolis, IN, USA

*To whom correspondence should be addressed.
[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors
Associate Editor: Ziv Bar-Joseph

Received on March 5, 2016; revised on September 3, 2016; accepted on September 30, 2016

Abstract

Motivation: Biclustering is widely used to identify co-expressed genes under subsets of all the conditions in a large-scale transcriptomic dataset. The program, QUBIC, is recognized as one of the most efficient and effective biclustering methods for biological data interpretation. However, its availability is limited to a C implementation and to a low-throughput web interface.
Results: An R implementation of QUBIC is presented here with two unique features: (i) a 82% average improved efficiency by refactoring and optimizing the source C code of QUBIC; and (ii) a set of comprehensive functions to facilitate biclustering-based biological studies, including the qualitative representation (discretization) of expression data, query-based biclustering, bicluster expanding, biclusters comparison, heatmap visualization of any identified biclusters and co-expression networks elucidation.
Availability and Implementation: The package is implemented in R (as of version 3.3) and is available from Bioconductor at the URL: http://bioconductor.org/packages/QUBIC, where installation and usage instructions can be found.
Contact: qin.ma@sdstate.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Advances in high-throughput technologies accelerated the generation of massive quantities of gene expression data. This data revolution is only partially paralleled by the development of new algorithms for its interpretation. Biclustering is a widely accepted approach for gene co-expression analysis to identify co-expressed genes under subsets of all the conditions in a gene expression dataset. Several biclustering algorithms such as Plaid (Lazzeroni and Owen, 2002), SAMBA (Tanay *et al.*, 2002), FABIA (Hochreiter *et al.*, 2010) have been published in the past two decades. It is noteworthy that our program, QUBIC (Li *et al.*, 2009) is reviewed as one of the best programs due to its prediction performance on benchmark datasets and as the best in real biological dataset tests (Eren *et al.*, 2013). To enable the biclustering users lacking comprehensive computational background, a web server of QUBIC was developed in 2012 (Zhou *et al.*, 2012). Since gene expression datasets keep increasing in scale, we developed this user requested R package of QUBIC (QUBIC-R for short), to provide an efficient optimized implementation and to eliminate large-scale data submission to a webserver.

The unique features of QUBIC-R include: (i) biclustering is integrated with analyses functions, i.e. data discretization, query-based biclustering, bicluster expanding, biclusters comparison, heatmap

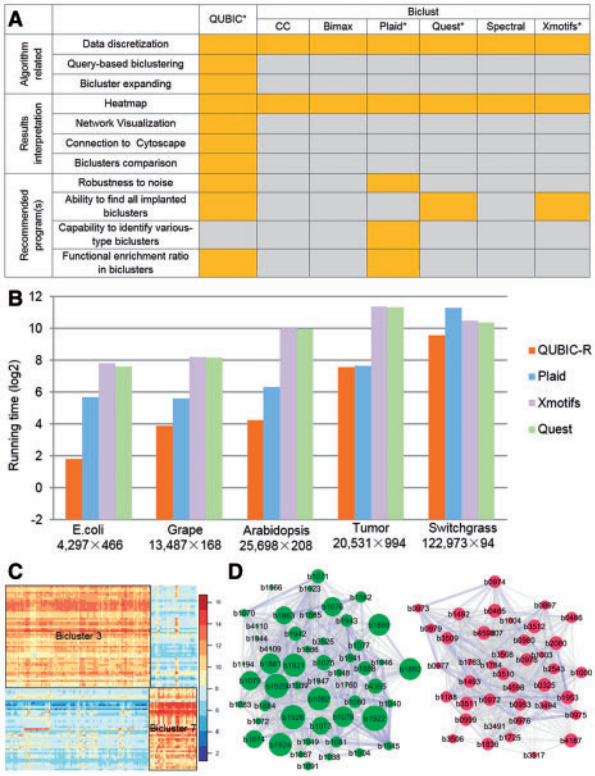

Fig. 1. (**A**) Comparison of QUBIC-R and 6 R packages in *biclust*. The yellow color indicates that a package provides the corresponding function or is recommended in a specific biclustering application and gray color represents the opposite; (**B**) comparison of running time among four recommended programs, annotated with asterisks in Figure 1A; (**C**) Heatmap visualization of two biclusters identified in *E. coli* data; (**D**) Co-expression networks of Figure 1C biclusters. Green nodes represent bicluster #3 and red nodes represent bicluster #7. The larger the size of a node, the higher its degree of presence; and the thicker an edge the greater its co-expression value is

visualization and co-expression network elucidation (Fig. 1A); (ii) the QUBIC source code is optimized and converted from GNU C to C++, thus has better memory control and is more efficient than the original QUBIC (Li *et al.*, 2009) (an average 82.4% savings in running time, Supplementary Table S1); (iii) on five large-scale datasets, QUBIC-R consistently performs the best among four popular tools according to the running time (Fig. 1B).

2 Implementation

QUBIC-R package is developed for the R statistical computing environment, and is released on Bioconductor (Gentleman *et al.*, 2004). It depends on the *biclust* package developed by Kaiser *et al.* (2009) to be compatible with the *biclust* output. Its output format can also be used by network analysis software, such as Cytoscape (Smoot *et al.*, 2011).

The original QUBIC program, written in GNU C with POSIX library, is limited in its portability. A memory leak may occur if the primary functions are called more than once. This problem was addressed by refactoring the C source code and transforming it into C++. Specifically, to avoid memory leak, we changed the majority of data structures and replaced C pointers by STL containers. We also optimized core function structures to facilitate future package updates and developments. The program efficiency has been significantly increased with the same predicting results (Fig. 1A). An input data as large as 30 000 × 30 000 can be finished within half an hour (detailed limits test are in Supplementary Fig. S1). All the computational experiments were conducted on a computer with Windows 7 × 64, Memory 48G, Intel Core i7-6700 3.4G.

3 Functions and examples

Nine functions are included in QUBIC-R. (i) *qudiscretize* creates a discrete matrix for a given matrix, i.e. the qualitative representation of input gene expression data; (ii) *BCQU* and (iii) *BCQUD* perform biclustering for continuous and discretized gene expression data, respectively; (iv) *query-based biclustering* allows users to input additional biological information to guide the biclustering progress (Method S1); (v) *bicluster expanding* expands existing biclusters under specified consistency level (Method S2); (vi) *biclusters comparison* compares biclusters obtained via different algorithms or parameters; (vii) *quheatmap* draws heatmap for any single or two predicted bicluster(s); (viii) *qunetwork* creates co-expression networks based on the identified biclusters (Method S3) and (ix) *qunet2xml* converts the constructed networks into XGMML format for further analysis in Cytoscape, Biomax and JNets. We use the genome-scale gene expression data collected under 466 conditions of *E. coli* (Faith *et al.*, 2008) as an example to illustrate how these functions work. An installation of R package is required (Example S1). Details of the *E. coli* example are in Example S2 and synthetic data and yeast expression data are in Example S3–4.

i. *qudiscretize* is useful to obtain discrete gene expression matrix. This matrix can be used in other biclustering program, where -1 represents lowly express, 0 represents normally express, and 1 represents highly express. For example:

```
> matrix1 <- ecoli[1:3,1:4]
> matrix1
> matrix2 <- qudiscretize(matrix1)
> matrix2
```

ii. *BCQU* and (iii) *BCQUD* are used as the biclustering method from package 'biclust', for example:

```
> res <- biclust (x = ecoli, method = BCQU(), f = 0.25)
> res1 <- biclust (x = qudiscretize(ecoli), method = BCQUD(), f = 0.25)
```

And QUBIC algorithm can be called independently via *qubiclust* and *qubiclust_d* for continous and discrete data, respectively (res, res1, res2 and res3 are identical):

```
> res2 <- qubiclust (x = ecoli, f = 0.25)
> res3 <- qubiclust_d (x = qudiscretize(ecoli), f = 0.25)
```

iv. Using the parameter *weight*, a user can conduct a query-based biclustering, with additional biological information.

```
> file = 511145.protein.links.v10.txt
> graph = read.graph(file, format = ncol)
> get.edgelist(graph, names = TRUE)
> E(graph)$weight
> weight<-get.adjacency(graph, attr = weight)
> res4 <- biclust (x = ecoli, method = BCQU(), weight = weight, f = 0.25)
```

v. Using the *seedbicluster* parameter, a user can expand existing biclustering results to recruite more genes according to certain consistency level:

```
> res5 <- biclust (x = ecoli, method = BCQU(), seedbicluster = res, f = 0.25)
> summary (res)
> summary (res5)
```

vi. Using the parameter *showinfo*, the biclustering results from different algorithms or from a same algorithm with different combinations of parameter can be compared:

```
> test <- ecoli [1:50,]
```

> *res6 <- biclust (test, method = BCQU(), verbose = F)*
> *res7 <- biclust (test, method = BCCC())*
> *res8 <- biclust (test, method = BCBimax())*
> *showinfo (test, c(res6, res7, res8))*

vii. We can visualize the identified biclusters using heatmap in support of overall expression pattern analysis, either for a single bicluster or for two biclusters (Fig. 1C):

> *par (mar = c(5, 4, 3, 5), cex.lab = 1.1, cex.axis = 0.5, cex.main = 1.1)*
> *quheatmap (ecoli, res, number = 4)*
> *par (mar = c(5, 4, 3, 5), cex.lab = 1.1, cex.axis = 0.5, cex.main = 1.1)*
> *quheatmap (ecoli, res, number = c(3, 7))*

viii. We can construct and visualize network for the identified biclusters, using the function *qunetwork*, either for a single bicluster or for two biclusters:

> *library(qgraph)*
> *net1<- qunetwork (ecoli, res, number = 4, group = 4, method-* = ""spearman"")
> *qgraph(net1[[1]], groups = net1[[2]], layout =* ""spring"", minimum = 0.6,
color = cbind(rainbow(length(net1[[2]]) - 1), ""gray"", edge.label = FALSE)
> *net2<- qunetwork (ecoli, res, number = c(3, 7), group = c(3, 7), method =* ""spearman"")
> *qgraph(net2[[1]], groups = net2[[2]], legend.cex = 0.5, layout =* ""spring"", *minimum = 0.6, color = c*("red", "blue", "gold", "gray"), *edge.label = FALSE)*

ix. The function *qunet2xml* can convert the constructed networks into XGMML format, facilitating further functional enrichment analysis (e.g. DAVID) and advanced network visualization (e.g. Cytoscape, Fig. 1D):

> *sink*("tempnetworkresult.gr")
> *qunet2xml (net2, minimum = 0.6, color = c*("red", "blue", "gray"))
> *sink ()*

4 Conclusion

Biclustering algorithms facilitate researchers in identification of co-expressed gene subsets in their gene expression dataset, and has become a useful approach for the interpretation of gene expression profile data. Our R package implements a well-cited biclustering algorithm, QUBIC. It provides more efficient source code and fully integrated functions to identify and analyze biclusters and visualize identified biclusters and corresponding co-expression networks. This package is a powerful tool for gene expression data mining and co-expression network modeling.

Funding

This work was supported by the State of South Dakota Research Innovation Center, the Agriculture Experiment Station of South Dakota State University, National Science Foundation of United States(0604755 and 1546869) and the National Natural Science Foundation of China (61402194).

Conflict of Interest: none declared.

References

Eren,K. *et al*. (2013) A comparative analysis of biclustering algorithms for gene expression data. *Brief. Bioinf*., **14**, 279–292.
Faith,J.J. *et al*. (2008) Many microbe microarrays database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Res*., **36**, D866–D870.
Gentleman,R.C. *et al*. (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*., **5**, R80.
Hochreiter,S. *et al*. (2010) FABIA: factor analysis for bicluster acquisition. *Bioinformatics*, **26**, 1520–1527.
Kaiser,S. *et al*. (2009) biclust: Bicluster algorithms. *R package version 0.7*. p. 2.
Lazzeroni,L. and Owen,A. (2002) Plaid models for gene expression data. *Stat. Sin*., **12**, 61–86.
Li,G. *et al*. (2009) QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Res*., **37**, e101.
Smoot,M.E. *et al*. (2011) Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, **27**, 431–432.
Tanay,A. *et al*. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18**, S136–S144.
Zhou,F. *et al*. QServer: a biclustering server for prediction and assessment of co-expressed gene clusters. 2012.