Biclique: An R package for Maximal Biclique Enumeration in Bipartite Graphs

Yuping Lu (yupinglu89@gmail.com)
University of Tennessee https://orcid.org/0000-0002-8113-7652
Charles A. Phillips
University of Tennessee
Michael A. Langston
University of Tennessee

Research note

Keywords: Biclique, Bipartite graph, Graph algorithms, Maximality, R package

Posted Date: February 12th, 2020

DOI: https://doi.org/10.21203/rs.2.16755/v2

Version of Record: A version of this preprint was published at BMC Research Notes on February 21st, 2020. See the published version at https://doi.org/10.1186/s13104-020-04955-0.

Abstract

Objective Bipartite graphs are widely used to model relationships between pairs of heterogeneous data types. Maximal bicliques are foundational structures in such graphs, and their enumeration is an important task in systems biology, epidemiology and many other problem domains. Thus, there is a need for an efficient, general purpose, publicly available tool to enumerate maximal bicliques in bipartite graphs. The statistical programming language R is a logical choice for such a tool, but until now no R package has existed for this purpose. Our objective is to provide such a package, so that the research community can more easily perform this computationally demanding task. Results Biclique is an R package that takes as input a bipartite graph and produces a listing of all maximal bicliques in this graph. Input and output formats are straightforward, with examples provided both in this paper and in the package documentation. Biclique employs a state-of-the-art algorithm previously developed for basic research in functional genomics. This package, along with its source code and reference manual, are freely available from the CRAN public repository at https://cran.r-project.org/web/packages/biclique/index.html .

Introduction

All graphs we consider are finite, simple, unweighted and undirected. They are also *bipartite*, which means their vertices can be partitioned into two *partite sets* so that the endpoints of each edge lie in different sets. In such a graph, a *biclique* is a complete bipartite subgraph, that is, a subgraph in which every subgraph vertex in one partite set is adjacent to every subgraph vertex in the other partite set. A biclique with p vertices in one partite set and q vertices in the other is denoted by $K_{p,q}$. A biclique is *maximum* if it is of largest size, with size measured by either its number of vertices (vertex-maximum) or its number of edges (edge-maximum). Finding a vertex-maximum biclique is *NP*-hard [1], while identifying an edge-maximum biclique can be accomplished in polynomial time [2]. A biclique is *maximal* if no vertex can be added to it to form a larger biclique.

The problem of enumerating all maximal bicliques has found utility in a host of applications. In the biological sciences, for example, it has been used for biclustering microarray data [3-5], modeling proteome-transcriptome relationships [6], identifying discriminating genotype patterns [7], optimizing phylogenetic tree reconstructions [8], discovering epidemiological patterns [9], identifying common gene-set associations [10], and integrating heterogeneous functional genomics data [11]. This problem is difficult in large part due to its combinatorial nature. A bipartite graph with n vertices may contain as many as $2^{n/2}$ maximal bicliques [12].

In previous work [13], we presented a fast, general-purpose algorithm for this task. We dubbed it the Maximal Biclique Enumeration Algorithm, MBEA, and presented along with it an improved version we termed iMBEA. In this paper, we describe a publicly available implementation of both algorithms wrapped in R [14]. Simply called *Biclique*, this R package invokes efficient implementations of MBEA and iMBEA

written in C. Our goal is to provide the scientific community with a practical, convenient and efficient tool for finding all maximal bicliques in bipartite graphs.

Methods

Implementation

Biclique consists of four R functions. The core function, *bi.clique*, invokes an efficient algorithm to enumerate maximal bicliques. Three utility functions, *bi.format*, *bi.print*, and *bi.degree*, provide formatting and output support.

The *bi.clique* function takes five arguments, four of which have default values. These five are: an input file name, an input file format (either an edge list (the default) or a binary matrix), two arguments, one for each partite set, that specify the minimum number of vertices required for a maximal biclique to be reported (the default is 3), and an argument specifying the algorithm to use, either MBEA or iMBEA (the default is iMBEA). Pseudocode for MBEA and iMBEA is shown in Algorithm 1. Because iMBEA differs from MBEA by only a handful of additional steps, the two algorithms are presented jointly, with starred lines denoting the steps unique to iMBEA. On dense graphs, iMBEA will usually be the faster algorithm, while on sparse graphs, both algorithms are apt to take about the same amount of time. We therefore recommend the use of iMBEA in most cases. See [13] for a thorough discussion of the two methods.

The three utility functions operate as follows. The *bi.print* function generates a visual histogram of the distribution of sizes of the maximal bicliques enumerated by the most recent call to *bi.clique*. The *bi.format* function augments a list of edges with a header line declaring the number of vertices and edges the list contains, as is required by *bi.clique*. The *bi.degree* function reads a bipartite graph and outputs the degree of each vertex.

Application

Biclique is invoked in R as follows:

bicliques = bi.clique(filename, left_least, right_least, version, filetype)

This function generates a list of bicliques, which in the above example are assigned to the *bicliques* variable. The *filename* argument is the name of the input file. Using “left” to denote the first partite set and “right” to denote the second, the *left_least* and *right_least* arguments specify the minimum number of vertices required from each respective partite set in order for a maximal biclique to be reported. The *version* argument specifies whether to use MBEA or iMBEA.

The *filetype* argument can be a little more complicated. It specifies the input file format, which must be either an edge list (0) or a binary matrix (1). The default value is edge list. Such a list is tab-separated, with the first line declaring the number of vertices in each partite set, followed by the number of edges in the graph. Each subsequent line contains a pair of text labels for an edge, with the edge’s left endpoint

listed first and its right endpoint second. The binary matrix format is also tab-separated. Example input files are provided with the package.

A sample bipartite graph is depicted in Figure 1, where vertices u_1, u_2, u_3, u_4 and u_5 are in the left partite set, while v_1, v_2, v_3 and v_4 are in the right. This graph is encoded as graph.el, shown in Table 1.

5	4	10
u1	v1	
u1	v2	
u1	v4	
u2	v1	
u2	v2	
u2	v4	
u3	v3	
u3	v4	
u4	v4	
u5	v4	

Table 1. The encoding of graph.el, stored in edge list format.

The use of *bi.clique* is exemplified in Sample Invocation 1, where *graph.el* denotes the sample graph just illustrated and encoded. Since neither *left_least* nor *right_least* is specified, all maximal bicliques with at least one edge will be reported. Similarly, since no v*ersion* argument is declared, iMBEA will be invoked by default. And since no *filetype* argument is provided, *graph.el* is assumed to be in edge list format. Summary information returned by *bi.clique* comprises a listing of the input's biclique distribution, its total number of bicliques, and its vertex- and edge-maximum biclique sizes.

```
> bicliques = bi.clique("graph.el", , , ,)
Biclique    Number
K5,1        1
K1,2        1
K2,3        1
Number of bicliques             : 3
Vertex-maximum biclique     : K5,1
Edge-maximum biclique       : K2,3
```

Sample Invocation 1. A basic call to *bi.clique*.

Biclique is available on CRAN at https://cran.r-project.org/web/packages/biclique/index.html. Included is an R-style reference manual with detailed descriptions of all arguments and options. This stable, CRAN-ready version can be installed in R with the command *install.packages("biclique")*. The latest version of *Biclique* can be obtained via *devtools::install_github("YupingLu/biclique")*. Questions or bugs can be submitted to the GitHub webpage. Included in the package are several example bipartite graphs, most of which we obtained from the Koblenz Network Connection [15].

Tests

All tests were conducted on a Dell server with an Intel Xeon E3-1220 v5 3.0GHz processor under the Red Hat Enterprise Linux 7 operating system, with 16GB DDR4 SDRAM, using. R 3.4.2. C code compiled with gcc 4.8.5. Eight bipartite graphs obtained from [15] were studied. As shown in Table 2, timings on them ranged from 0.005 seconds to 21.094 seconds. These tests were not meant to be comprehensive, but instead merely to demonstrate that this software can handle affiliation graphs, authorship graphs, interaction graphs and others in addition to the various biological and random graphs tested in [13].

Graph	Left	Right	Edges	Bicliques	Vertex-Max	Edge-Max	Timings
S. African Companies	6	5	13	8	K4,1	K3,2	0.005
Southern Women 2	5	5	14	12	K3,1	K2,2	0.005
Southern Women 1	18	14	89	63	K14,1	K5,4	0.007
Club Membership	25	15	95	60	K21,1	K11,2	0.006
Corporate Leadership	20	24	99	66	K12,1	K9,2	0.007
American Revolution	136	5	160	14	K59,1	K59,1	0.006
Crime	829	551	1476	620	K1,25	K1,25	0.035
arXiv cond-mat	16726	22015	58595	21905	K1,116	K1,116	21.094

Table 2. Timings on eight sample bipartite graphs.

Conclusions

Biclique provides convenient access, through R, to cutting-edge algorithms for maximal biclique enumeration in bipartite graphs. It provides users with a means to extract relationships between pairs of heterogeneous entities, without a need to worry about implementations of complex codes such as MBEA/iMBEA. Biclique also produces extremal information, including the sizes of vertex-maximum and edge-maximum bicliques. *Biclique* has been tested on a variety of graphs, and is available on both CRAN and GitHub.

Availability and Requirements

Project name: *Biclique*.

Project home page: https://github.com/YupingLu/biclique

Operating system(s): Platform independent.

Programming language: R.

Other requirements: R version 3.4.0 or later is recommended.

License: GNU General Public License version 2.0 (GPL-2).

Any restrictions to use by non-academics: None.

Limitations

Biclique enumeration can be output bound. The number of bicliques in large, dense graphs can exceed machine memory limitations.

Abbreviations

MBEA: Maximal Biclique Enumeration Algorithm

iMBEA: Improved Maximal Biclique Enumeration Algorithm

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and material

Data used in this study are available at the Koblenz Network Collection (http://konect.uni-koblenz.de/)

Competing interests

The authors declare that they have no competing interests.

Funding

This research has been supported in part by the National Institutes of Health under grant R01AA018776 and by the Environmental Protection Agency under grant G17D112354237. These funding agencies had no role in the design of the study, in the collection, analysis, and interpretation of data, or in writing the manuscript.

Authors' contributions

YL implemented the package and performed testing. CAP led exhaustive software evaluations. MAL directed and supported the research. All authors assisted in the preparation, reading and final approval of this typescript.

Acknowledgements

Not applicable.

References

1. Peeters R: **The maximum edge biclique problem is NP-complete**. *Discrete Applied Mathematics* 2003, **131**(3):651-654.
2. Garey MR, Johnson DS: **Computers and intractability: a guide to the theory of NP-completeness**: W. H. Freeman and Company; 1979.
3. Cheng Y, Church GM: **Biclustering of expression data**. In: *Proceedings, International Conference on Intelligent Systems for Molecular Biology: 2000*. 93-103.
4. Tanay A, Sharan R, Shamir R: **Discovering Statistically Significant Biclusters in Gene Expression Data**. *Bioinformatics* 2002, **18**:136-144.
5. Wang H, Wang W, Yang J, Yu PS: **Clustering by pattern similarity in large data sets**. In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data; Madison, Wisconsin*. 564737: ACM 2002: 394-405.
6. Kirova R, Langston MA, Peng X, Perkins AD, Chesler EJ: **A Systems Genetic Analysis of Chronic Fatigue Syndrome: Combinatorial Data Integration from SNPs to Differential Diagnosis of Disease**. In: *Methods of Micorarray Data Analysis VI.* Edited by McConnell P, Lim S, Cuticchia AJ. Scotts Valley, California: CreateSpace Publishing; 2009: 81-98.
7. Yosef N, Yakhini Z, Tsalenko A, Kristensen V, Børresen-Dale A-L, Ruppin E, Sharan R: **A supervised approach for identifying discriminating genotype patterns and its application to breast cancer data**. *Bioinformatics* 2007, **23**(2):e91-e98.
8. Sanderson MJ, Driskell AC, Ree RH, Eulenstein O, Langley S: **Obtaining maximal concatenated phylogenetic data sets from large sequence databases**. *Mol Biol Evol* 2003, **20**(7):1036-1042.
9. Mushlin RA, Kershenbaum A, Gallagher ST, Rebbeck TR: **A graph-theoretical approach for pattern discovery in epidemiological research**. *IBM Systems Journal* 2007, **46**(1):135-149.
10. Chesler EJ, Langston MA: **Combinatorial Genetic Regulatory Network Analysis Tools for High Throughput Transcriptomic Data**. In: *Systems Biology and Regulatory Genomics.* Edited by Eskin E, vol. 4023: Springer; 2006: 150–165.
11. Baker EJ, Jay JJ, Philip VM, Zhang Y, Li Z, Kirova R, Langston MA, Chesler EJ: **Ontological discovery environment: a system for integrating gene-phenotype associations**. *Genomics* 2009, **94**(6):377-387.
12. Prisner E: **Bicliques in graphs I: bounds on their number**. *Combinatorica* 2000, **20**(1):109-117.
13. Zhang Y, Phillips CA, Rogers GL, Baker EJ, Chesler EJ, Langston MA: **On Finding Bicliques in Bipartite Graphs: a Novel Algorithm and Its Application to the Integration of Diverse Biological Data Types**. *BMC Bioinformatics* 2014, **15**(110).
14. Team RC: **R: a language and environment for statistical computing**. In. Vienna, Austria: R Foundation for Statistical Computing; 2017.
15. Kunegis J: **KONECT: the Koblenz network collection**. In: *Proceedings of the 22nd International Conference on World Wide Web; Rio de Janeiro, Brazil*. 2488173: ACM 2013: 1343-1350.

Figures

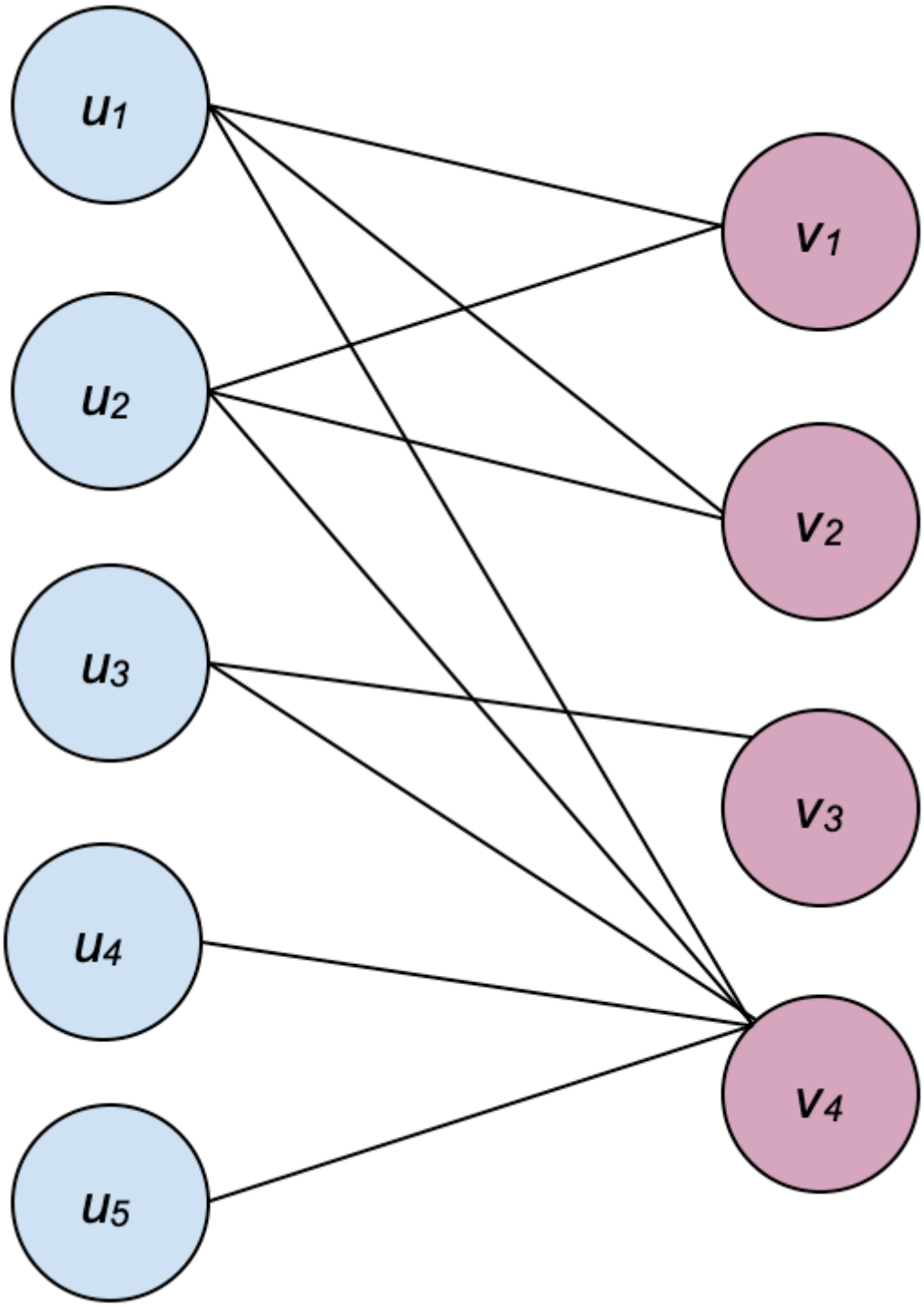

Figure 1

A sample bipartite graph.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Alogrithm1.pdf