Bioinformatics, 35(17), 2019, 3181–3183
doi: 10.1093/bioinformatics/btz027
Advance Access Publication Date: 14 January 2019
Applications Note

Gene expression

EBIC: an open source software for high-dimensional and big data analyses

Patryk Orzechowski[1,2,*] **and Jason H. Moore**[1,*]

[1]Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104, USA and [2]Department of Automatics and Robotics, AGH University of Science and Technology, Krakow 30-059, Poland

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on August 14, 2018; revised on November 17, 2018; editorial decision on December 30, 2018; accepted on January 12, 2019

Abstract

Motivation: In this paper, we present an open source package with the latest release of Evolutionary-based BIClustering (EBIC), a next-generation biclustering algorithm for mining genetic data. The major contribution of this paper is adding a full support for multiple graphics processing units (GPUs) support, which makes it possible to run efficiently large genomic data mining analyses. Multiple enhancements to the first release of the algorithm include integration with R and Bioconductor, and an option to exclude missing values from the analysis.
Results: Evolutionary-based BIClustering was applied to datasets of different sizes, including a large DNA methylation dataset with 436 444 rows. For the largest dataset we observed over 6.6-fold speedup in computation time on a cluster of eight GPUs compared to running the method on a single GPU. This proves high scalability of the method.
Availability and implementation: The latest version of EBIC could be downloaded from http://github.com/EpistasisLab/ebic. Installation and usage instructions are also available online.
Contact: patryk.orzechowski@gmail.com or jhmoore@upenn.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Biclustering is an unsupervised machine learning technique which attempts to detect meaningful data patterns that are distributed across different columns and rows of the input dataset. This allows biclustering to capture heterogeneous patterns that manifest only in subsets of genes and subsets of samples. Biclustering has been commonly applied to genomic datasets (Padilha and Campello, 2017) and has proven to be successful in revealing potential diagnostic biomarkers (Liu *et al.*, 2018), and claimed to identify altered transcriptional profiles in breast cancer (Singh *et al.*, 2018).

With exponentially increasing sizes of the input datasets, there is an emerging need for effective and efficient methods that would scale well with growing amounts of data. Although there was discussion on possibility of applying biclustering to larger datasets (Kasim *et al.*, 2016; Padilha and Campello, 2017), hardly any biclustering study involved large genomic dataset. This motivated the emergence of parallel biclustering method. Some of the most recent parallel methods use multiple threads—e.g. runibic (Orzechowski *et al.*, 2017), or Message Passing Interface—e.g. ParBiBit (González-Domínguez and Expósito, 2018), or graphics processing unit (GPU)—e.g. CCS (Bhattacharya and Cui, 2017).

One of the recent advancements in biclustering area was introduction of Evolutionary-based BIClustering (EBIC), which takes advantage of multiple other evolutionary computation strategies (Orzechowski *et al.*, 2018a, b). This representative of hybrid biclustering algorithms (Orzechowski and Boryczko, 2016a, b, c) has been shown to outperform multiple state-of-the-art methods in terms of accuracy. Although the original concept of EBIC provided theoretical support for multiple GPUs, all the previous evaluations have been made using a single GPU. Thus, the rationale of involving multiple GPUs was not clear. Another constraint for EBIC was hardware limitation of the size of the dataset to 65 535 rows per GPU. This required large clusters of GPUs in order to run analyses and greatly restricted application of the method.

© The Author(s) 2019. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Downloaded from https://academic.oup.com/bioinformatics/article/35/17/3181/5289325 by guest on 20 April 2024

In this paper, we introduce the open source package built on top of the upgraded version of the method. First and foremost, a full support for multi-GPUs is added, which allows to analyze datasets with almost unlimited numbers of rows (available memory of devices are the only constraint). Second, an integration with Bioconductor was added, which enables the user to run all the analysis from the R level. Third, a different method for performing analysis was added, which depends on the presence or absence of missing values within the data. Last, but not least, some bugs have been fixed and optimizations were made for more efficient memory management. All above combined make this open source software ready out-of-the-box or big data biclustering analyses.

2 Materials and methods

The major objective of EBIC is to make data analyses as easy as possible. Thus, in its simplest form, the method requires only a single parameter, which is a dataset in a popular comma-separated values format. Running the algorithm on multiple GPUs requires to specify additional parameters with number of required devices. An example commands to run the software are as follows (the second one runs the method on 4 GPUs):

```
./ebic -i input.txt
./ebic -i input.txt -g 4
```

After a proper installation, the method may also be run within R environment, using *system* function, which points to the location of EBIC binary. For detailed information on command line parameters, and general information on software installation and usage please refer to Supplementary Material as well as online documentation available on Github.

The new version of EBIC introduced herein provides a comprehensive open source framework for performing biclustering analysis. The major improvements over the original release of the method include:

- Support for Big Data. In the previous version of EBIC only a very limited number of rows could be processed on a single GPU. Kernel grid constrained the maximal number of rows analyzable by an EBIC to 65 535 per GPU. Thus, at least eight GPUs were needed to analyze large datasets, e.g. modern methylation datasets. Our new implementation overcomes this limitation, allowing to analyze up to 2^{31}–1 rows per single GPU (devices with computing capabilities 3.0). This greatly enhances the flexibility and applicability of the method to almost any type of data. This comes at a cost of reducing the size of genetic algorithm population down to 65 535. This remains a large number, as for the majority of genomic datasets the algorithm converged using a population size of 1600 or less individuals.
- Handling missing values. We introduce a very important feature which allows to remove the impact of missing values on the results of the method. As EBIC search is driven by counting of rows, a greater or equal relation between the values in columns used to capture missing values, instead of the real trends in the data. This posed a drawback, especially for datasets with high percentage of missing values. Instead of finding useful patterns in the data, EBIC used to become more attracted in detecting the emptiness. In the current release, missing values might be replaced with a predefined value (e.g. 0 or 999) passed as an input parameter. This value no longer counted toward the score of the bicluster. Thus, the method is more focused on detecting the trends, instead of emptiness. Please be aware that performing a dataset normalization could impact the representation of missing values in each column if missing values are involved in calculations.
- Different input file formats support. EBIC allows different delimiters in input file, which simplifies portability of data between R and EBIC. The input datasets needs to meet the following restrictions: (i) the data values might be separated by either comma, tabulator, space or semicolon, (ii) the labels of rows and columns need to be present (in left-most column and top-most row) and (iii) the left- and top-most field representing crossing of columns and rows labels needs not to be empty.
- Compatibility with R and Bioconductor. The results returned by EBIC could be easily saved into a format loadable by Bioconductor R package *biclust* in order to perform biological validation. In Supplementary Material we provide detailed workflow presenting how to use EBIC, all within R environment. Notice that EBIC is not a part of Bioconductor and it needs to be installed separately.
- Workflow for methylation data analysis. EBIC was capable to capture bio-meaningful signals in methylation data. A tutorial is presented in a Supplementary Material.

3 Results

In order to assess running times of the algorithm we have performed tests on from one up to eight GPUs on datasets with varying number of rows and columns. The Gene Expression Omnibus accession numbers of the datasets as well as run times of the algorithm are presented in Table 1.

EBIC obtained up to 6.6× fold speedup using eight GPUs on a dataset with over 436 k rows. For datasets with smaller number of rows, the speedups were around 1.2× (12 k rows) and 2.75× (55 k rows). The relation between the different number of GPUs used and obtained speedup is presented in Figure 1.

Table 1. Datasets used in the experiment as well as an average running time (in minutes) using a cluster of eight GeForce GTX 1080 Ti GPUs

Dataset	Genes	Samples	Description	Run time (min)
GDS1490	12 483	150	Neural tissue profiling	7.1
GSE65194	54 675	178	Breast cancer	18.3
GSE84493	436 444	310	Prostate cancer methylation	24.5

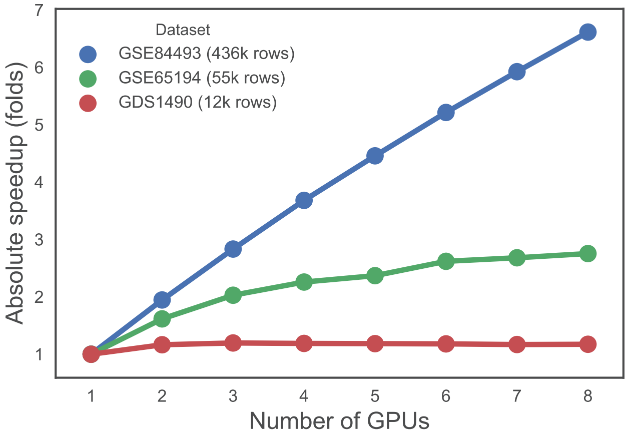

Fig. 1. Speedups obtained using multiple GPUs (GeForce GTX 1080 Ti) for the datasets from Table 1

Downloaded from https://academic.oup.com/bioinformatics/article/35/17/3181/5289325 by guest on 20 April 2024

Since its proposal EBIC was applied to numerous datasets: both synthetic and real genomic ones. It was found to be very robust in detecting correlated genomic patterns. EBIC was found to be especially helpful in detecting large biclusters, in which multiple rows are correlated with each other within some subset of columns. We believe that EBIC could serve as a useful tool for performing exploratory analysis of correlations of rows within the data.

4 Conclusions

In this paper, we present the recent advancements in one of the leading biclustering methods. The algorithm was wrapped into a framework, which is conveniently integrated with R and allows multiple input file formats. In Supplementary Material we also demonstrate that even for such a large genomic dataset, the results provided by EBIC are bio-meaningful. We conclude that EBIC, released as open source package, is a very convenient tool for getting insight from large genomic datasets.

Future work on EBIC involves developing a better initialization method of the initial population, to make a method more robust for wide biclusters. We also consider integrating expert knowledge into a software. As running EBIC on some of the datasets may result in highly overlapping biclusters, we also consider modifications to the list storing the best found solutions in order to limit the overlap between the biclusters.

Funding

This research was supported in part by PL-Grid Infrastructure and by grants [LM012601, TR001263, ES013508] from the National Institutes of Health (USA).

Conflict of Interest: none declared.

References

Bhattacharya,A. and Cui,Y. (2017) A GPU-accelerated algorithm for biclustering analysis and detection of condition-dependent coexpression network modules. *Sci. Rep.*, **7**, 4162.

González-Domínguez,J. and Expósito,R.R. (2018) ParBiBit: parallel tool for binary biclustering on modern distributed-memory systems. *PLoS One*, **13**, e0194361.

Kasim,A. *et al.* (2016) *Applied Biclustering Methods for Big and High-Dimensional Data Using R*. CRC Press, Boca Raton, FL.

Liu,Y.-C. *et al.* (2018) Biclustering of transcriptome sequencing data reveals human tissue-specific circular RNAs. *BMC Genomics*, **19**, 958.

Orzechowski,P. and Boryczko,K. (2016a) Hybrid biclustering algorithms for data mining. In: Squillero,G. and Burelli,P. (eds). *Applications of Evolutionary Computation*. Springer International Publishing, Cham, pp. 156–168.

Orzechowski,P. and Boryczko,K. (2016b) Propagation-based biclustering algorithm for extracting inclusion-maximal motifs. *Comput. Inform.*, **35**, 391–410.

Orzechowski,P. and Boryczko,K. (2016c) Text mining with hybrid biclustering algorithms. In: Rutkowski,L. *et al.* (eds). *Artificial Intelligence and Soft Computing*. Springer International Publishing, Cham, pp. 102–113.

Orzechowski,P. *et al.* (2017) runibic: a Bioconductor package for parallel row-based biclustering of gene expression data. *Bioinformatics*, **34**, 4302–4304.

Orzechowski,P. *et al.* (2018a) *Ebic: a next-generation evolutionary-based parallel biclustering method*. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. pp. 59–60. ACM, New York, NY, USA.

Orzechowski,P. *et al.* (2018b) EBIC: an evolutionary-based parallel biclustering algorithm for pattern discovery. *Bioinformatics*, **34**, 3719–3726.

Padilha,V.A. and Campello,R.J. (2017) A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, **18**, 55.

Singh,A. *et al.* (2018) TuBA: tunable biclustering algorithm reveals clinically relevant tumor transcriptional profiles in breast cancer. *bioRxiv*, 245712.

Downloaded from https://academic.oup.com/bioinformatics/article/35/17/3181/5289325 by guest on 20 April 2024