EBIC: a scalable biclustering method for large scale data analysis

Patryk Orzechowski*†
University of Pennsylvania
Philadelphia, PA 19104, USA
patryk.orzechowski@gmail.com

Jason H. Moore
University of Pennsylvania
Philadelphia, PA 19104
jhmoore@upenn.edu

ABSTRACT

Biclustering is a technique that looks for patterns hidden in some columns and some rows of the input data. Evolutionary search-based biclustering (EBIC) is probably the first biclustering method that combines high accuracy of detection of multiple patterns with support for big data. EBIC has been recently extended to a multi-GPU method and allows to analyze very large datasets. In this short paper, we discuss the scalability of EBIC as well as its suitability for RNA-seq and single cell RNA-seq (scRNA-seq) experiments.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Cluster analysis**; *Search methodologies*; Bio-inspired approaches; • **Theory of computation** → *Massively parallel algorithms*;

KEYWORDS

biclustering, data mining, unsupervised machine learning, evolutionary computation, genetic programming

ACM Reference Format:
Patryk Orzechowski and Jason H. Moore. 2019. EBIC: a scalable biclustering method for large scale data analysis. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3326762

1 INTRODUCTION

Biclustering, an unsupervised machine learning technique, has recently caught a lot of traction in bioinformatics [18]. Unlike traditional clustering methods, biclustering methods can capture local patterns, called biclusters, which are present only in selected rows, and in selected columns. This might be one of the reasons of rising popularity of those dedicated approaches in biomedical sciences, as certain phenomena (e.g. regulatory mechanisms) could be identified only in certain subsets of patients with a given disease.

A recent advance in biclustering was the development of Evolutionary search-based Biclustering (EBIC). This method was shown to capture multiple patterns [6] in data (such as column-based, row-based, monotonous, constant, up-regulated, shift-, scale- and shift-scale) with average accuracy high over 90% and achieved highest proportion of significantly enriched biclusters for the benchmark collection of datasets from Eren et al. [3] among multiple established biclustering methods [14].

In our recent paper, the original method was scaled into a multi-GPU algorithm, which can handle practically any size of the dataset within the hardware constraints [12]. The latest release of the software was equipped with very useful tools for data analytics. Integration with a popular open-source platform for bioinformatics Bioconductor, as well as support for missing (or empty) value was among the most important features for this open source software.

In this paper, we summarize major accomplishments of the method and confirm obtained speedups.

2 METHODS

EBIC is a modern multi-GPU biclustering algorithm, which is based on evolutionary search in column space. The method uses good practices of GPU programming [7] and takes from the previous experience of the authors in developing GPU-based biclustering methods [8]. At each of the iteration of the method, the new population of biclusters is created by performing simple genetic operations. Each bicluster is represented by a series of columns, which imposes a monotonous order of the values in rows of the bicluster. The number and indices of row of a bicluster are not known upfront. Verification of monotonous order of values in each of the rows performed in parallel on a cluster of GPUs allows the method to determine the rows. The fitness function of EBIC encourages large biclusters, the columns are slightly more preferable than the rows. This trick drives evolution towards acquiring new columns.

Simple genetic operations are used for creating a series of columns in EBIC. The operators include adding a new column, changing the value of an existing column, changing the positions of two columns, removing a column from series, or performing a cross-over operation, in which two series are combined with each other.

EBIC belongs to a family of hybrid biclustering methods [9–11], as it combines selected concepts of different methods. For example, the concept of using monotonous trends for pattern detection is a technique adapted from OPSM [1] and later successfully adapted in two powerful biclustering algorithms: UniBic and its R parallel equivalent runibic [13, 17]. In order to minimize data transfers between CPU and GPU, a Compressed Bicluster Format (CBF) was proposed as the representation of biclusters. This format was inspired by Compressed Row Storage (CRS), a popular representation for sparse matrices [2] and uses two arrays: first, in which the elements indicate the position at which a bicluster given by index starts, and the second, in which the actual column identifiers of the

*corresponding author
†Patryk Orzechowski is also affiliated with Department of Automatics and Robotics, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326762

biclusters are stored. This representation allows EBIC to analyze at each iteration a given number biclusters with any length of the columns of each bicluster. The mechanism for enhancing the diversity of the solutions is based on a crowding technique [16], as each appearance of a column in a population increases the suppression rate for this column. EBIC uses elitism [15] and imports a certain number of best solutions to the next population. Finally, EBIC uses a tabu list [4, 5] in order to forbid reanalysis of the same bicluster.

3 RESULTS

In order to verify the scalability of the method, we generated a toy datasets with shift-scale pattern with 100 columns and a varying number of rows (10k, 100k and 1M). EBIC was run on each of those datasets with different number of GPUs, what caused a different number of rows to be dispatched across the GPUs. We have used a cluster with four GeForce GTX 1080 devices. The running times on a single GPU for the dataset with 10k rows, 100k rows, and 1M rows were 3.5 mins, 20.7 mins, and 141.1 mins respectively. Obtained relative speedups were presented in Fig. 1.

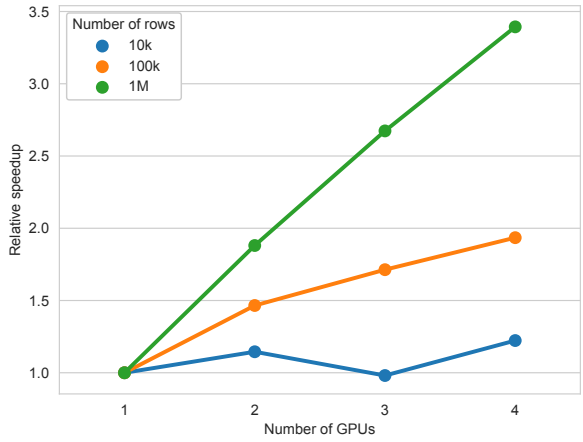

Figure 1: Relative speedup for a dataset with 100 columns and different number of rows for different number of GPUs compared to the running time on a single GPU.

We have observed a speedup of 3.4 folds on 4 GPUs for the datasets with 1M rows and 1.9 folds for the dataset with 100k rows. For the dataset with 10k rows, hardly any benefit for using more than a single device was observable. Using three devices deteriorated the performance, as not enough data was available to cover data transfer overheads.

4 CONCLUSIONS

With multi-GPU support and improved scalability, EBIC allows to analyze large scale data of almost any size. The method is efficient in detection of large representations of correlated rows under subsets of columns. This is a highly desirable property in bioinformatics, especially in genomics, as correlated genes may take part in same biological processes. EBIC can also detect negative and approximate patterns and has some tolerance to noise and outliers. Integration with Bioconductor allows the method to provide an easily integrable environment and can become a part of R pipelines for analyzing large genomic datasets. Setting a custom representation of missing value changes the course of evolution and makes EBIC focus on non-empty regions, instead of extracting large areas filled with zeroes. This aspect is very helpful for analyzing RNA-seq or single cell RNA-seq (scRNA-seq) data, as they usually contain multiple zeroes. Overall, EBIC is a very useful tool built on top of evolutionary algorithm which offers novel insight into the data.

ACKNOWLEDGMENTS

This research was supported in part by PLGrid Infrastructure and by NIH grants LM010098, LM012601 and AI116794.

REFERENCES

[1] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. 2003. Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.* 10, 3-4 (2003), 373–384.
[2] Aydin Buluç, Jeremy T Fineman, Matteo Frigo, John R Gilbert, and Charles E Leiserson. 2009. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*. ACM, 233–244.
[3] Kemal Eren, Mehmet Deveci, Onur Küçüktunç, and Ümit V Çatalyürek. 2013. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics* 14, 3 (2013), 279–292.
[4] Fred Glover. 1989. Tabu search - part I. *ORSA Journal on computing* 1, 3 (1989), 190–206.
[5] Fred Glover. 1990. Tabu search - part II. *ORSA Journal on computing* 2, 1 (1990), 4–32.
[6] S. C. Madeira and A. L. Oliveira. 2004. Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 1, 1 (2004), 24–45.
[7] Patryk Orzechowski and Krzysztof Boryczko. 2015. Effective biclustering on GPU-capabilities and constraints. *Prz Elektrotechniczn* 1 (2015), 133–6.
[8] Patryk Orzechowski and Krzysztof Boryczko. 2015. Rough assessment of GPU capabilities for parallel PCC-based biclustering method applied to microarray data sets. *Bio-Algorithms and Med-Systems* 11, 4 (2015), 243–248.
[9] Patryk Orzechowski and Krzysztof Boryczko. 2016. Hybrid Biclustering Algorithms for Data Mining. In *Applications of Evolutionary Computation*, Giovanni Squillero and Paolo Burelli (Eds.). Springer International Publishing, Cham, 156–168.
[10] Patryk Orzechowski and Krzysztof Boryczko. 2016. Propagation-Based Biclustering Algorithm for extracting inclusion-maximal motifs. *Computing & Informatics* 35, 2 (2016).
[11] Patryk Orzechowski and Krzysztof Boryczko. 2016. Text Mining with Hybrid Biclustering Algorithms. In *Artificial Intelligence and Soft Computing*, Leszek Rutkowski, Marcin Korytkowski, Rafał Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada (Eds.). Springer International Publishing, Cham, 102–113.
[12] Patryk Orzechowski and Jason H. Moore. 2019. EBIC: an open source software for high-dimensional and big data analyses. *Bioinformatics* (2019), btz027. https://doi.org/10.1093/bioinformatics/btz027
[13] Patryk Orzechowski, Artur Pańszczyk, Xiuzhen Huang, and Jason H Moore. 2018. runibic: a Bioconductor package for parallel row-based biclustering of gene expression data. *Bioinformatics* 34, 24 (2018), 4302–4304. https://doi.org/10.1093/bioinformatics/bty512
[14] Patryk Orzechowski, Moshe Sipper, and Xiuzhen Huang. 2018. EBIC: an evolutionary-based parallel biclustering algorithm for pattern discovery. *Bioinformatics* 34, 21 (05 2018), 3719–3726. https://doi.org/10.1093/bioinformatics/bty401
[15] Riccardo Poli, William B Langdon, Nicholas F McPhee, and John R Koza. 2008. *A field guide to genetic programming*. Lulu.com.
[16] Bruno Sareni and Laurent Krahenbuhl. 1998. Fitness sharing and niching methods revisited. *IEEE transactions on Evolutionary Computation* 2, 3 (1998), 97–106.
[17] Zhenjia Wang, Guojun Li, Robert W Robinson, and Xiuzhen Huang. 2016. UniBic: Sequential row-based biclustering algorithm for analysis of gene expression data. *Scientific reports* 6 (2016).
[18] Juan Xie, Anjun Ma, Anne Fennell, Qin Ma, and Jing Zhao. 2018. It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. *Briefings in bioinformatics* (2018). https://doi.org/10.1093/bib/bby014