Rapid Prototyping of Evolution-Driven Biclustering Methods in Julia

Paweł Renc
AGH University of Science and Technology
30-059 Krakow, Poland
rencpawe@gmail.com

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

Aleksander Byrski
AGH University of Science and Technology
30-059 Krakow, Poland
olekb@agh.edu.pl

Jarosław Wąs
AGH University of Science and Technology
30-059 Krakow, Poland
jarek@agh.edu.pl

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

ABSTRACT

Biclustering is a technique of detecting meaningful patterns in tabular data. It is also one of the fields in which evolutionary algorithms have risen to the very top in terms of speed and accuracy. In this short paper we summarize the results of porting one of the leading evolutionary-based biclustering methods EBIC to Julia - an emerging high-end programming language. This is probably the first biclustering package developed in this programming language. The main findings of this study are that flexibility combined with high performance make Julia an appealing platform for development and validation of new biclustering methods.

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, machine learning, evolutionary computation, parallel algorithms

ACM Reference Format:
Paweł Renc, Patryk Orzechowski, Aleksander Byrski, Jarosław Wąs, and Jason H. Moore. 2021. Rapid Prototyping of Evolution-Driven Biclustering Methods in Julia. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3462739

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

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '21 Companion, July 10–14, 2021, Lille, France
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3462739

1 INTRODUCTION

Biclustering aims at finding one different types of patterns manifested in subsets of rows and columns [2, 3]. The major application of this technique is biomedical field, or more specifically gene expression analysis. In recent years evolutionary biclustering approaches have become among the most accurate biclustering methods, pushing the research towards new levels.

One of the obstacles slowing development of novel methods has been lack of highly productive programming language that would allow rapid development and validation of prototypes in just a few lines of code. Julia, an emerging high level programming language that combines the power of expression of Python with efficiency and scalability of C++, seemed to be a promising lead.

In this short paper we analyze the performance of EBIC.jl – an implementation of well-established evolutionary biclustering method EBIC in Julia, introduced in [8]. We evaluate the new method on a large collection of datasets as well as also share our perspective on a future adoption of Julia as a platform for biclustering.

2 METHODS

The following leading biclustering methods have been included in our analysis:
Runibic [5], a parallel version of UniBic [9], captures the longest common monotonous trends between different pairs of rows. Those sequences are later expanded first to strict and later to approximate order-preserving biclusters.
RecBic [1] is an exhaustive greedy biclustering method focused on expanding monotonous patterns. The method initially evaluates combinations of 3 randomly selected columns, which are further expanded until convergence is reached.
EBIC [4, 6] is a multi-GPU evolutionary biclustering method that looks for monotonously increasing trends. Combinations of columns determined with simple genetic operators (insertion, deletion, substitution, swap, or crossover) are evaluated in parallel on GPU. EBIC incorporates tournament selection, elitism, crowding and tabu list.
EBIC.jl introduced in [8] is a novel implementation of EBIC in Julia. The minor improvements involved optimizing GPU kernel by loop unrolling, using atomic operators, as well as further parameter

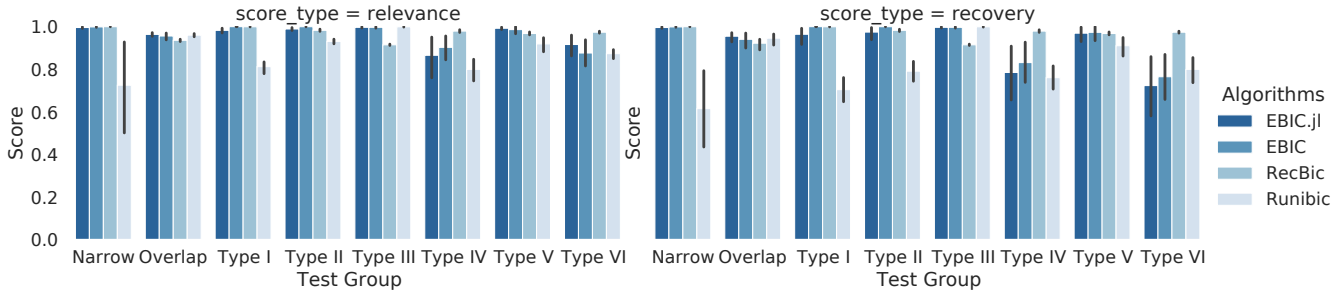

Figure 1: Relevance (left) and recovery (right) of biclustering methods on the collection from Wang et al.

tuning that resulted in reducing running time of the method. EBIC.jl at this point can be run on a single GPU only.

All the aforementioned methods were benchmarked with their default settings on the collection of 119 datasets organized in 8 different problem groups from Wang et al. [9]. Their performance was evaluated on tasks of detecting: narrow, overlapping, trend-preserving (Type I), column-constant (Type II), row-constant (Type III), shift (Type IV), scale (Type V), and shift-scale (Type VI) patterns. The measures used for comparison were two popular metrics in biclustering called recovery and relevance, which were proposed by Prelic et al. [7]. The first one captured how well a method is able to find ground truth biclusters, whereas the second reflected the similarity of the generated biclusters to the ground truth.

3 RESULTS

The performance of the newly proposed EBIC.jl was found to be highly competitive with the leading methods in the field, yielding the best results for Overlap and Type V scenarios (Figure 1). EBIC.jl visibly underperformed in Type IV scenario, with both recovery and relevance scores lower than EBIC and RecBic. For Type VI scenarios Ebic.jl shows higher relevance than EBIC, but at cost of lower recovery. Both methods were outperformed there by RecBic.

In terms of the running times, the implemented optimizations allowed EBIC.jl to converge much faster than EBIC for the vast majority of the datasets. Nonetheless, this comparison can't be deemed fully objective, as both EBIC and EBIC.jl feature different GPU kernels. Runibic and RecBic were the fastest in the comparison, however it needs to be noted, that the datasets had very low size (up to 1000 rows), which handicaped the performance of GPU methods.

4 DISCUSSION

In this short paper we have presented the results of evaluation of newly developed implementation of one of the leading biclustering methods, EBIC, in Julia. This is probably the first or certainly one of the first biclustering packages developed in this language. The method has been shown to be very competitive with the leading biclustering methods in the field.

The second major contribution of this paper is practical verification of feasibility of using Julia as a programming language for biclustering. We have noticed notable benefits, as high level paradigm allowed to greatly expedite the development process. Needless to say, the source code has been reduced by half and made clearer and more manageable. The new version of the method also simplifies rapid validation of new concepts and allows to refocus future efforts on rapid prototyping of new solutions.

This study proves that Julia has already become an appealing programming language to work on solving complex machine learning problems, including biclustering. All combined, if popularized, Julia has potential of becoming a 'go-to' platform for biclustering.

ALGORITHM AVAILABILITY

EBIC.jl is available here: https://github.com/EpistasisLab/EBIC.jl

ACKNOWLEDGMENTS

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

REFERENCES

[1] Xiangyu Liu, Di Li, Juntao Liu, Zhengchang Su, and Guojun Li. 2020. RecBic: a fast and accurate algorithm recognizing trend-preserving biclusters. *Bioinformatics* 36, 20 (07 2020), 5054–5060. https://doi.org/10.1093/bioinformatics/btaa630

[2] 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.

[3] Patryk Orzechowski, Krzysztof Boryczko, and Jason H Moore. 2019. Scalable biclustering — the future of big data exploration? *GigaScience* 8, 7 (06 2019). https://doi.org/10.1093/gigascience/giz078 giz078.

[4] Patryk Orzechowski and Jason H Moore. 2019. EBIC: an open source software for high-dimensional and big data analyses. *Bioinformatics* 35, 17 (01 2019), 3181–3183. https://doi.org/10.1093/bioinformatics/btz027 arXiv:https://academic.oup.com/bioinformatics/article-pdf/35/17/3181/29591797/btz027.pdf

[5] 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

[6] 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

[7] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22, 9 (2006), 1122–1129.

[8] Paweł Renc, Patryk Orzechowski, Aleksander Byrski, Jarosław Wąs, and Jason H. Moore. 2021. EBIC.JL - an Efficient Implementation of Evolutionary Biclustering Algorithm in Julia. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3449726.3463197

[9] 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, 1 (2016), 1–10.