SoftwareX 26 (2024) 101666

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

bioScience: A new python science library for high-performance computing bioinformatics analytics

Aurelio López-Fernández [a], Francisco A. Gómez-Vela [a,*], Jorge Gonzalez-Dominguez [b], Parameshachari Bidare-Divakarachari [c]

[a] Intelligent Data Analysis Group (DATAi), Universidad Pablo de Olavide, ES-41013, Seville, Spain
[b] Computer Architecture Group, CITIC, Universidade da Coruña, Campus de Elviña, A Coruña, 15071, Spain
[c] Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India

ARTICLE INFO

Keywords:
Bioinformatics
High-performance computing
Data science
Data mining
Data analysis

ABSTRACT

BioScience is an advanced Python library designed to satisfy the growing data analysis needs in the field of bioinformatics by leveraging High-Performance Computing (HPC). This library encompasses a vast multitude of functionalities, from loading specialized gene expression datasets (microarrays, RNA-Seq, etc.) to preprocessing techniques and data mining algorithms suitable for this type of datasets. BioScience is distinguished by its capacity to manage large amounts of biological data, providing users with efficient and scalable tools for the analysis of genomic and transcriptomic data through the use of parallel architectures for clusters composed of CPUs and GPUs.

Code metadata

Current code version	v0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00803
Permanent link to Reproducible Capsule	N/A
Legal Code License	BSD 3-Clause License
Code versioning system used	GIT
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python ≥ 3.10 and pip ≥ 23.0
If available Link to developer documentation/manual	English user manuals
Support email for questions	alopfer1@upo.es

Software metadata

Current software version	v0.1
Permanent link to executables of this version	https://pypi.org/project/bioscience/
Permanent link to Reproducible Capsule	N/A
Legal Software License	BSD 3-Clause License
Computing platforms/Operating Systems	Unix, MacOS, Microsoft Windows
Installation requirements & dependencies	Python ≥ 3.10 and pip ≥ 23.0
If available, link to user manual - if formally published include a reference to the publication in the reference list	English user manuals
Support email for questions	alopfer1@upo.es

1. Motivation and significance

In the present era, there exists an unprecedented process of data generation and collection, primarily facilitated by advancements in information technologies [1]. This fact needs the adaptation of data analysis methodologies and computational approaches to handle the vast amount, diversity, speed, and authenticity of the data being currently produced, transmitted, and analyzed [2]. In this context, Machine Learning (ML) algorithms are utilized to discover significant,

* Corresponding author.
E-mail addresses: alopfer1@upo.es (Aurelio López-Fernández), fgomez@upo.es (Francisco A. Gómez-Vela).

https://doi.org/10.1016/j.softx.2024.101666
Received 29 November 2023; Received in revised form 14 February 2024; Accepted 16 February 2024
Available online 19 February 2024
2352-7110/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

complex, and practical patterns within datasets from diverse domains such as bioinformatics [3], social networks [4], energy consumption [5] among others.

Biclustering is a critical machine learning technique that is employed in the analysis of the aforementioned data [6]. These techniques are capable of grouping elements of a dataset based on the intrinsic similarity present in the data [7]. Specifically, Biclustering possesses the capability to discern local patterns by identifying instances exhibiting similar behavior according to a subset of their characteristics. For this reason, Biclustering is becoming an increasingly significant technique in the analysis of complex data, particularly genetic data [8]. Although these algorithms offer several benefits, they do possess a significant drawback: they pose an NP-Hard computational problem. As a result, the majority of these algorithms have implemented heuristics in order to enhance their computational efficiency. Consequently, the increase in the volume and quantity of datasets, mainly those containing gene expression data, is leading to an effort by the scientific community to improve the performance of these algorithms [9].

In this regard, the computational capabilities offered by High-Performance Computing (HPC) can help to address this challenge. Biclustering techniques, mainly focused on parallel and distributed computing, such as Apache Hadoop [10] or Spark [11] have offered good results in this context. Additionally, programming based on Graphics Processing Units (GPU) emerges as one of the main HPC techniques for intensive parallel data processing [12,13]. In this context, a considerable number of biclustering algorithms have utilized GPU technology to expedite result acquisition with notable success [14–16]. Nevertheless, the primary issue encountered by these approaches is their tendency to be excessively intricate for non-programming expert users, thereby limiting their impact within the scientific community. Consequently, there is a necessity for the implementation of biclustering algorithms that not only achieve commendable results within a reasonable time but are also accessible to the scientific community.

In this paper, we present bioScience, an advanced Python library designed to satisfy the growing data analysis needs in the field of bioinformatics by leveraging HPC. This library covers an extensive array of features, ranging from importing specific gene expression datasets (including microarrays, RNA-Seq, etc.) to preprocessing methods and different algorithms specifically designed for such datasets. BioScience stands out for its ability to handle large volumes of biological data, offering users effective and adaptable resources for the examination of genomic and transcriptomic data. This is achieved by harnessing parallel architectures that combine CPUs and GPUs within cluster systems. The library, accompanied by comprehensive documentation and examples of its execution, is accessible to the public for download at the following URL: https://pypi.org/project/bioscience/.

2. Software description

BioScience is a Python library that can be conveniently obtained through PyPI (https://pypi.org/project/bioscience) or cloned and executed as a Python script directly from GitHub (https://github.com/aureliolfdez/bioscience). It is open-source under the BSD 3-Clause license, allowing source and binary format use, redistribution, and modification on GitHub. Installable and executable from a terminal, it integrates easily into pipelines without Python knowledge. Alternatively, it can run in Google Colab, saving personal computing resources. BioScience aims to simplify data analysis for bioinformaticians, handling datasets of any size and processing needs.

2.1. Software architecture

As stated previously, bioScience is wholly developed in Python. As a consequence, the source code is structured into files and classes containing a succession of methods that implement functionality from the library. The library is composed of three primary packages, as illustrated in the package diagram (refer to Fig. 1). The *base package* comprises methods and classes utilized by the remaining packages; the *preprocess package* implements a sequence of general and specialized preprocessing techniques for biological datasets; and the *dataMining package* integrates algorithms for data mining. This package incorporates the BiBit Biclustering algorithm [17] within an environment optimized for HPC.

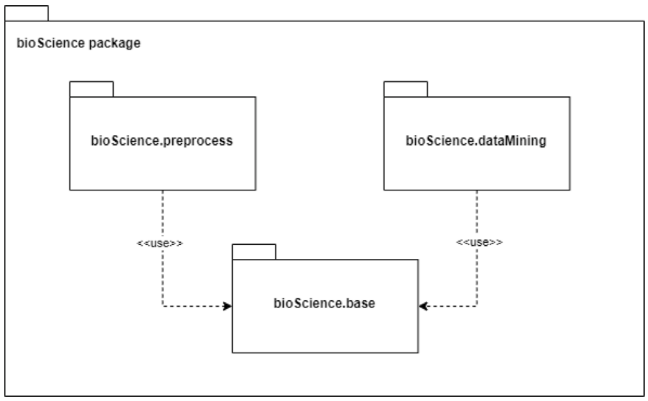

Fig. 1. Package diagram of bioScience.

Building upon the package structure of bioScience, Fig. 2 presents a global perspective of the internal structure encompassing every package in the library. In this figure, the individual classes, methods, and corresponding files for each of these packages are described in detail, emphasizing the modular structure of the bioScience library. This comprehensive analysis not only facilitates comprehension of the library's architecture but also provides developers and researchers with a roadmap for efficiently navigating and employing the library in the context of biological data analysis. The following is a description of each of the actors involved for each package:

bioScience.base.Files This file contains a function for uploading biological datasets, including unprocessed, preprocessed, and even data that has been externally binarized. Additionally, it supports a variety of sequencing technologies, including RNA-Seq and microarrays. Furthermore, it comprises a number of functions that enable the storage of data mining technique execution results in files. Ultimately, the user may also store binarized datasets if they have carried out a binarisation process in bioScience library.

bioScience.base.Models.Dataset The dataset concept is represented by this class within the library. Different categories of data, including the original dataset and the preprocessed/binarized dataset, are stored within a Dataset object. Furthermore, any additional pertinent data, including gene and column names, as well as supplementary information like gene length, that may be present in the RNA-Seq datasets, are also preserved.

bioScience.base.Models.BiclusterModel The purpose of this class is to store the results independently of the Biclustering algorithm executed. Furthermore, the execution time of debug executions will be recorded so that the behavior of every algorithm utilized in an HPC environment can be evaluated.

bioScience.base.Models.Bicluster A class that represents a bicluster has been developed by using the Biclustering techniques in this library. For this purpose, data such as the rows and columns involved and the original data of the elements involved in the bicluster are stored.

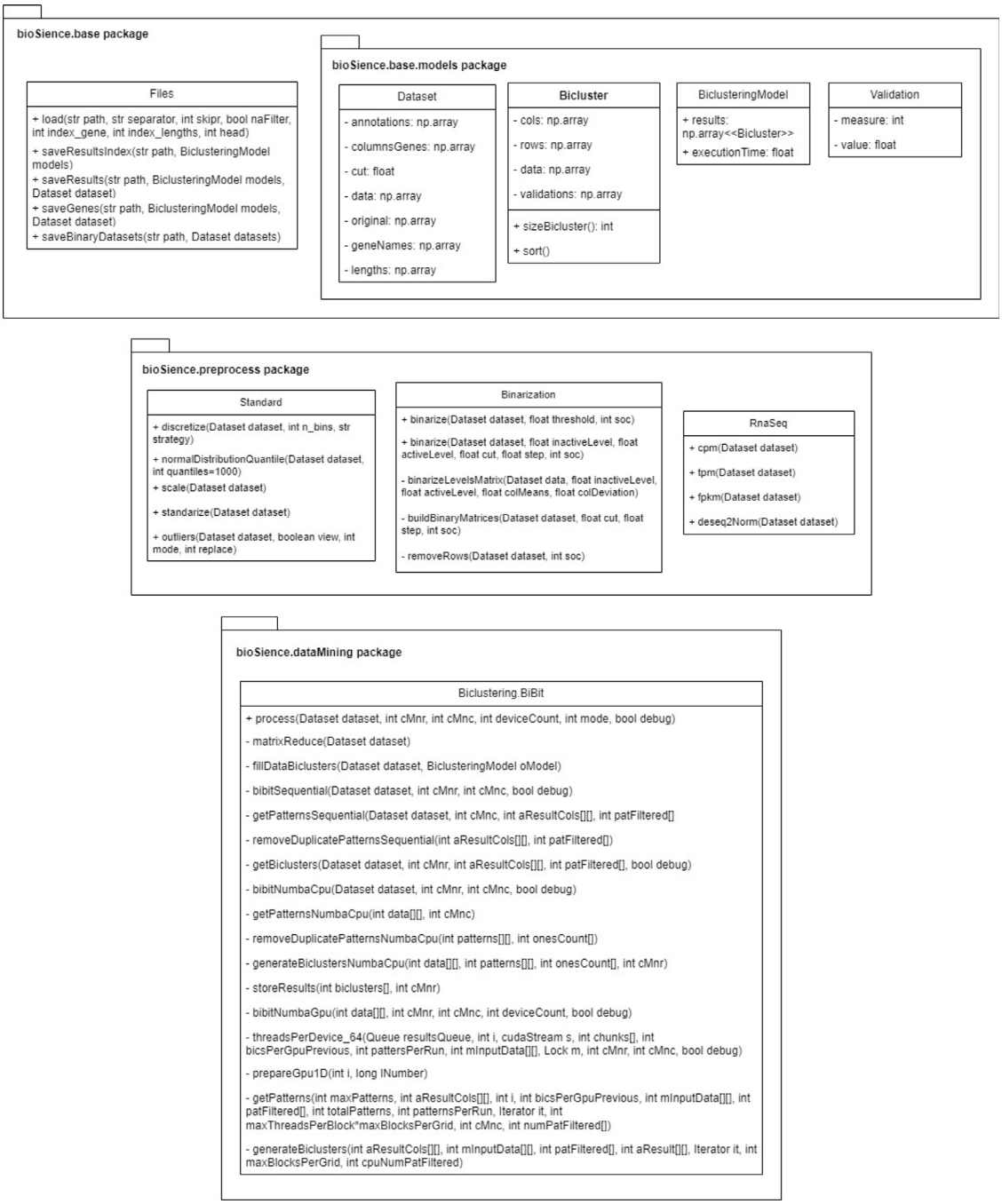

Fig. 2. Classes and files for each bioScience package.

bioScience.base.Models.Validation This class represents any validation measure that is used after any data mining technique has been executed.

bioScience.preprocess.Standard This file includes multiple basic preprocessing methods that can be used for any type of dataset such as discretization, standardization and normalization. Outlier detection and treatment using the interquartile range (IQR) is another method included.

bioScience.preprocess.Binarization bioScience provides the capability to perform a binarization transformation on the dataset, as certain data mining techniques, such as the BiBit Biclustering algorithm, may necessitate the dataset to be in binary format. In pursuit of this objective, the user is provided with the option to either binarize the dataset itself or generate multiple binary datasets via fuzzy logic. This functionality is implemented to mitigate any noise that might be introduced during the data transformation process.

bioScience.preprocess.RnaSeq This class incorporates preprocessing methods that are specifically used for RNA-Seq datasets, e.g. CPM (Counts Per Million), TPM (Transcripts Per Kilobase Million), FPKM (Fragments Per Kilobase Million) [18] or DESeq2 [19].

bioScience.dataMining.Biclustering.BiBit This file contains three versions of the BiBit algorithm: sequential, parallel for multiple CPUs and parallel for multi-GPUs architectures. These versions will be executed depending on the value provided by the user

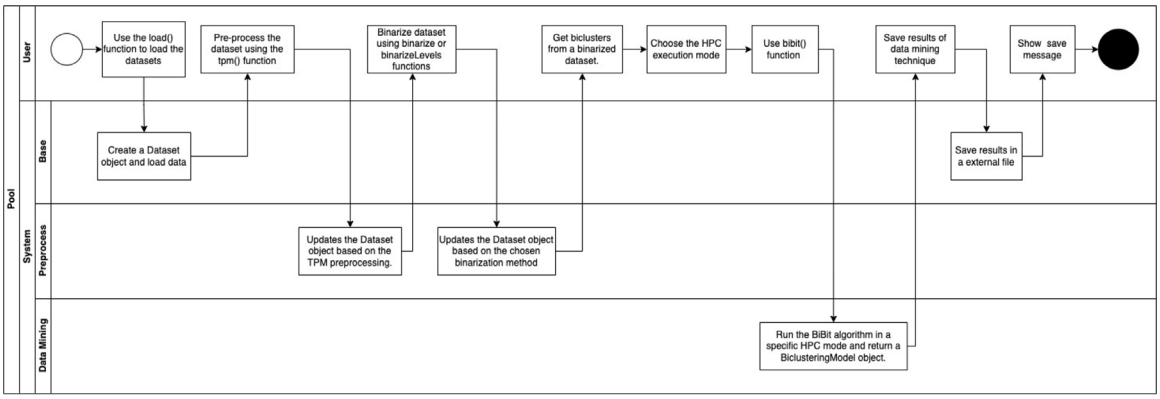

Fig. 3. Main use case business process.

in the mode attribute of the process function. If the value of this attribute is equal to 1, the BiBit algorithm will be executed sequentially, while if the value is equal to 2 it will be executed in a CPU parallel environment. On the other hand, if the value of this attribute is 3, this algorithm will run in a multi-GPU architecture. The development of this HPC environment has been supported by the use of the NUMBA library.

2.2. Software functionalities

BioScience provides specialized features for analyzing biological datasets in HPC environments, enhancing computational capabilities and broadening its use in intensive settings. Its functionalities include:

- Dataset Importation: BioScience allows simple uploading of diverse datasets, including synthetic datasets and those associated with sequencing technologies like microarrays or RNA-Seq. It also supports raw, preprocessed, or binarized data.
- High-Performance Computing (HPC): The library is optimized for HPC environments, enabling fast computational speeds and facilitating rigorous experiments.
- Preprocessing: BioScience offers various preprocessing options such as standardization, discretization, normalization, outlier detection and treatment, and specific techniques for RNA-Seq data like CPM, TPM, FPKM, and DESeq2.
- Binarization: Users can binarize datasets using simple or fuzzy logic methods to support certain data mining techniques that require binary representation.
- Adaptation to HPC: The library is designed to adapt data mining techniques to HPC environments, with support for parallel execution using CPU or multiple GPU devices.
- Algorithm Modification: The BiBit binary Biclustering algorithm has been modified for execution either sequentially or across multiple GPU devices.
- Usability and Transparency: Technical functionalities are embedded in the user experience to optimize computational performance, including support for large datasets, memory management, and resource utilization.
- Result Exportability: All results from data mining techniques, including binary datasets generated during binarization, are exportable.

To enhance comprehension of the primary functionalities of this module, a typical sequence of user–machine interactions from dataset loading to result export is depicted in Fig. 3. Assigning a system actor to each row delineates the user–machine interaction. The activities of the user are depicted in the first panel. The three primary components of the bioScience system, which consist of the library packages, are illustrated in the second, third, and fourth rows. The figure illustrates the primary use case, which describes the user's process of accessing the system and conducting an analysis on an RNA-Seq dataset. Initially, the dataset is loaded into the system by the user via the *load()* function. Subsequently, the user intends to apply TPM preprocessing to the data. Following this, in order to use the BiBit binary clustering algorithm, it is necessary to binarize the preprocessed dataset. After the Biclustering algorithm is executed subsequent to binarization, the outcomes are ultimately stored in an external file.

2.3. Memory management in bioscience

In High-Performance Computing (HPC) contexts, efficient memory management is essential due to the disparity between computation capacity, storage capacity, and GPU device memories [20]. Hence, it is imperative to make prudent choices about the utilization of these memories for efficient processing of extensive data sets, ensuring optimal performance and mitigating the risk of memory overflow issues.

Efficient data transfers between different memory units, such as RAM and GPU memory, and optimizing the use of available computing power and memory are crucial considerations. Therefore, bioScience employs a memory resource planning strategy with the aim of optimizing the utilization of all available resources at both the computational and storage levels.

When the environment exclusively utilizes CPU processors, the library divides the dataset into equal segments, taking into account the current amount of RAM available on the computer. This prevents the occurrence of memory overrun and enables the execution of larger datasets. In multi-GPU situations, bioScience can distribute chunks of the dataset evenly among the GPU devices in a cluster. The primary objective of these endeavors is to optimize the use of existing hardware resources and handle input datasets without encountering any memory overflow problems.

3. Illustrative examples

In order to demonstrate the functionality of bioScience, the objective of this section is to conduct an experiment using synthetic, microarray, RNA-Seq, and single-cell datasets. This experiment was conducted on a Amazon EC2 instance equipped with an Intel Xeon Platinum 8275L featuring 24 cores operating at 3 GHz, 61 GB of RAM, and 8 NVIDIA A100 16 GB graphics cards, each offering a combined total of 6912 CUDA cores. Table 1 presents an overview of the experiment, including the dataset sizes, whether any preprocessing was performed, the number of biclusters generated, and the time required to produce the results of data mining techniques. In addition, Fig. 4 presents four graphs in which the execution times of each type of execution for each dataset used are shown visually.

The first dataset is a synthetic binary matrix sized 4000 × 4000, with 15% of values uniformly set to 1. Since it is synthetic and binary,

Table 1
Summary of the experimentation including the time of execution.

Dataset	Genes	Features	Preprocess	Biclusters	HPC	Time (s)
Synthetic	4000	4000	None	7,998,000	None CPU GPU (1) GPU (2)	2017.03 170.75 19.93 9.65
Microarray	3016	6	Binarization	737,505	None CPU GPU (1) GPU (2)	12.49 3.98 0.45 0.21
RNA-Seq	27,179	12	TPM & Binarization	3596	None CPU GPU (1) GPU (2)	18 119.35 1513.45 163.12 80.65
Single-cell	22,593	533	Binarization	149	None CPU GPU (1) GPU (2)	15 320.43 1987.48 223.61 108.25

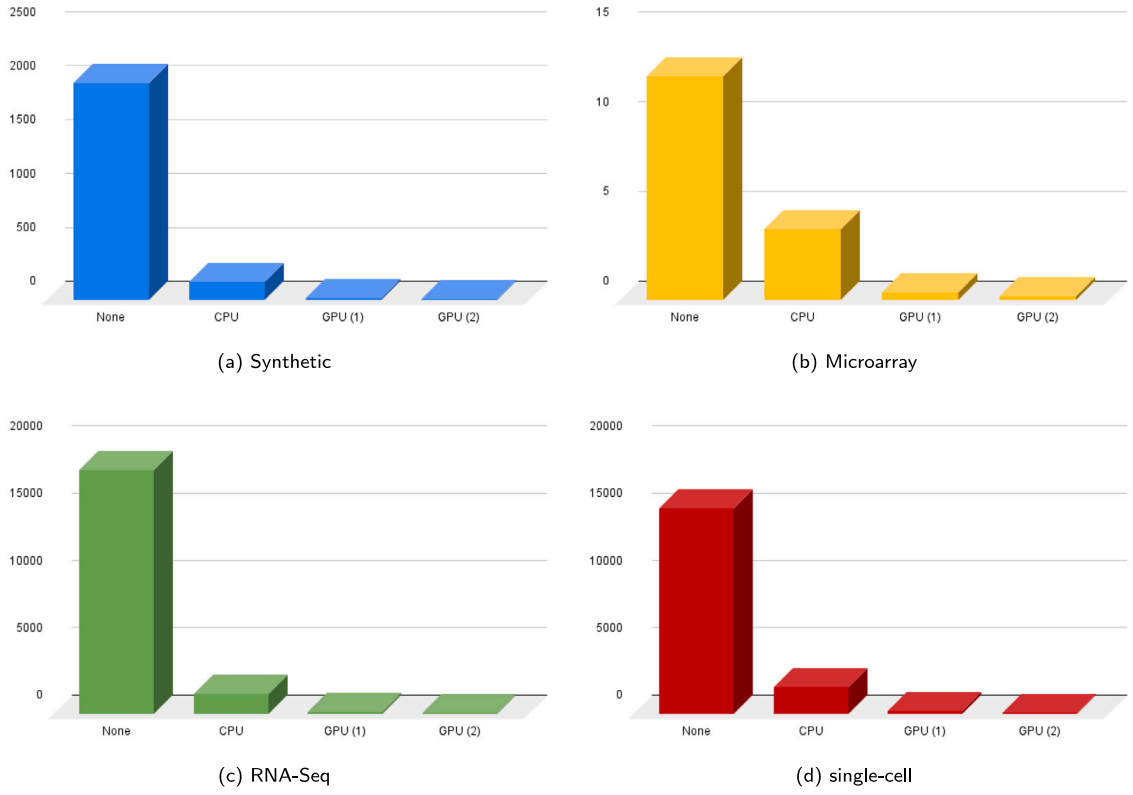

Fig. 4. Visual representation of the time of execution for the experimentation performed for each dataset.

no preprocessing is required. The dataset is loaded using the *load()* function, and the Biclustering algorithm is implemented with MNR and MNC values set to 2 using the *bibit()* function to obtain the maximum possible biclusters. Approximately 8 million biclusters are obtained, highlighting the need for an HPC environment, especially with GPU devices, for efficient execution of data mining techniques expected to yield extensive results, as indicated in the summary table (Table 1).

The microarray dataset (GSE26910) consists of 3016 genes showing differential expression in breast cancer across six samples. This illustrates the ability of data mining approaches to produce substantial results, even when working with smaller datasets. Before executing the data mining process, the dataset is transformed into binary format using the *binarize()* function. Despite the dataset's small size, factors such as processing and memory overhead, and under-utilization of computational resources in an HPC environment can limit computational performance [21]. Nonetheless, the use of HPC in this library results in decreased execution times, as shown in the summary table of this section.

The last two datasets have a higher volume, resulting in data mining algorithms needing to manage a significantly burdensome workload, irrespective of the volume of output they produce. The two datasets consist of RNA-Seq (GSE60450) [22] and single-cell (GSE246622) [23] data. The RNA-Seq dataset has a size of 27 179 × 19, while the single-cell dataset has a size of 22 593 × 533. The count matrix for the RNA-Seq dataset has been treated using the tpm() and binarize() methods. However, for the single-cell dataset, only binarizing has been applied as this dataset is already pre-processed. These two examples illustrate that as the amount and size of the dataset rises, the difference in performance between a sequential approach and a high-performance computing (HPC) environment becomes more noticeable.

4. Impact

Despite the existence of numerous libraries and tools that facilitate the analysis of bioinformatics data, data mining techniques are frequently confronted with escalating execution times for their outputs. This can be attributed to either the escalating volume of the datasets or the substantial workloads associated with them. Hence, in response to this requirement, the bioScience software is proposed, which incorporates data analysis methods and generic preprocessing techniques like discretization, normalization, standardization, handling outlier and binarization, in addition to RNA-Seq dataset-specific preprocessing methods.

Using the capabilities of HPC to expedite these bioinformatics data analyses is the primary benefit of bioScience. This is imperative in order to expedite scientific breakthroughs and apply their implications in clinical and research environments. The initial iteration of the library incorporates a data mining technique that is available in three variations: sequential, multi-CPU parallel, and GPU parallel. This feature enhances the library's adaptability to various hardware configurations, thereby accommodating the hardware specifications and availability of its users. For this reason, we believe that bioScience has great potential for widespread adoption among the Bioinformatics community.

Moreover, in order to improve the usability, decrease the programming effort, and increase the number of potential users, bioScience is easy to install with the `pip` package and includes both execution examples and detailed documentation (https://bioscience.readthedocs.io). Finally, bioScience is continuously evolving, and there are plans to increase the number and variety of preprocessing and data mining methods, as well as to include other modules such as validation and visualization. Therefore, in the future, the impact of bioinformatics analyses will be even greater.

5. Conclusions

BioScience, a Python library designed to analyze large bioinformatics datasets and whose methods can be executed on HPC hardware including multiple CPUs and GPUs, is introduced in this work. Presently, the library comprises a preprocessing module (incorporating both general methods and specific approaches tailored to RNA-Seq and single-cell data) and a data analytics module. In addition, for proper maintenance of the library, it is intended to further develop modules to facilitate validation and result visualization and to incorporate new methods for preprocessing and data analysis. Since the chosen techniques are extensively implemented in bioinformatics pipelines, bioScience can be utilized to expedite a vast array of analyses. Finally, the library will be tested and improved on upgraded hardware to incorporate new features from other types of GPUs from vendors like AMD or Intel.

CRediT authorship contribution statement

Aurelio López-Fernández: Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Conceptualization. **Francisco A. Gómez-Vela:** Writing – review & editing, Writing – original draft, Conceptualization. **Jorge Gonzalez-Dominguez:** Writing – review & editing, Writing – original draft, Conceptualization. **Parameshachari Bidare-Divakarachari:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used are publicly accessible and publicly available in verified databases. In addition, artificially generated data have been used.

Acknowledgment

Funding for open access publishing: Universidad Pablo de Olavide/CBUA

References

[1] Cozzoli Nicola, Salvatore Fiorella Pia, Faccilongo Nicola, Milone Michele. How can big data analytics be used for healthcare organization management? Literary framework and future research from a systematic review. BMC Health Serv Res 2022;22(1):1–14. http://dx.doi.org/10.1186/s12913-022-08167-z.

[2] Batko Kornelia, Ślęzak Andrzej. The use of big data analytics in healthcare. J Big Data 2022;9(1):3. http://dx.doi.org/10.1186/s40537-021-00553-4.

[3] Kashyap Hirak, Ahmed Hasin Afzal, Hoque Nazrul, Roy Swarup, Bhattacharyya Dhruba Kumar. Big data analytics in bioinformatics: architectures, techniques, tools and issues. In: Network modeling analysis in health informatics and bioinformatics. Vol. 5, Springer; 2016, p. 1–28. http://dx.doi.org/10.1007/s13721-016-0135-4.

[4] Chaudhary Kiran, Alam Mansaf, Al-Rakhami Mabrook S, Gumaei Abdu. Machine learning-based mathematical modelling for prediction of social media consumer behavior using big data analytics. J Big Data 2021;8(1):1–20. http://dx.doi.org/10.1186/s40537-021-00466-2.

[5] Mostafa Noha, Ramadan Haitham Saad Mohamed, Elfarouk Omar. Renewable energy management in smart grids by using big data analytics and machine learning. Mach Learn Appl 2022;9:100363. http://dx.doi.org/10.1016/j.mlwa.2022.100363.

[6] José-García Adán, Jacques Julie, Sobanski Vincent, Dhaenens Clarisse. Biclustering algorithms based on metaheuristics: a review. In: Metaheuristics for machine learning: new advances and tools. Springer; 2022, p. 39–71. http://dx.doi.org/10.1007/978-981-19-3888-7_2.

[7] Madeira Sara C, Oliveira Arlindo L. Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 2004;1(1):24–45. http://dx.doi.org/10.1109/TCBB.2004.2.

[8] Xie Juan, Ma Anjun, Fennell Anne, Ma Qin, Zhao Jing. It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. Brief Bioinform 2019;20(4):1450–65. http://dx.doi.org/10.1093/bib/bby014.

[9] Orzechowski Patryk, Boryczko Krzysztof, Moore Jason H. Scalable biclustering—the future of big data exploration? GigaScience 2019;8(7):giz078. http://dx.doi.org/10.1093/gigascience/giz078.

[10] Lin Qin, Xue Yun, Chen Wensheng, Ye Shuqun, Li Wanli, Liu Jingjing. Parallel large average submatrices biclustering based on MapReduce. In: 2015 11th international conference on computational intelligence and security. CIS, IEEE; 2015, p. 134–7. http://dx.doi.org/10.1109/CIS.2015.40.

[11] Lin Qin, Zhang Huailing, Wang Xizhao, Xue Yun, Liu Hongxin, Gong Changwei. A novel parallel biclustering approach and its application to identify and segment highly profitable telecom customers. IEEE Access 2019;7:28696–711. http://dx.doi.org/10.1109/ACCESS.2019.2898644.

[12] Dafir Zineb, Lamari Yasmine, Slaoui Said Chah. A survey on parallel clustering algorithms for big data. Artif Intell Rev 2021;54:2411–43. http://dx.doi.org/10.1007/s10462-020-09918-2.

[13] López-Fernández Aurelio, Rodríguez-Baena Domingo S, Gómez-Vela Francisco. gMSR: A multi-GPU algorithm to accelerate a massive validation of biclusters. Electronics 2020;9(11):1782. http://dx.doi.org/10.3390/electronics9111782.

[14] Kakati P, Bhattacharyya DK, Kalita Jugal K. BicBioEC: biclustering in biomarker identification for ESCC. In: Network modeling analysis in health informatics and bioinformatics. Vol. 8, Springer; 2019, p. 1–21. http://dx.doi.org/10.1007/s13721-019-0200-x.

[15] Orzechowski Patryk, Moore Jason H. EBIC: an open source software for high-dimensional and big data analyses. Bioinformatics 2019;35(17):3181–3. http://dx.doi.org/10.1093/bioinformatics/btz027.

[16] Bhattacharya Anindya, Cui Yan. A GPU-accelerated algorithm for biclustering analysis and detection of condition-dependent coexpression network modules. Sci Rep 2017;7(1):4162. http://dx.doi.org/10.1038/s41598-017-04070-4.

[17] Rodriguez-Baena Domingo S, Perez-Pulido Antonio J, Aguilar-Ruiz Jesus S. A biclustering algorithm for extracting bit-patterns from binary datasets. Bioinformatics 2011;27(19):2738–45. http://dx.doi.org/10.1093/bioinformatics/btr464.

[18] Chatterjee Aniruddha, Ahn Antonio, Rodger Euan J, Stockwell Peter A, Eccles Michael R. A guide for designing and analyzing RNA-seq data. In: Gene expression analysis: methods and protocols. Springer; 2018, p. 35–80. http://dx.doi.org/10.1007/978-1-4939-7834-2_3.

[19] Love Michael I, Huber Wolfgang, Anders Simon. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 2014;15(12):1–21. http://dx.doi.org/10.1186/s13059-014-0550-8.

[20] Wang Qihan, Peng Zhen, Ren Bin, Chen Jie, Edwards Robert G. MemHC: An optimized GPU memory management framework for accelerating many-body correlation. ACM Trans Archit Code Optim 2022;19(2). http://dx.doi.org/10.1145/3506705.

[21] Finn Michael P, Liu Yan, Mattli David M, Behzad Babak, Yamamoto Kristina H, Guan Qingfeng, Shook Eric, Padmanabhan Anand, Stramel Michael, Wang Shaowen. High-performance small-scale raster map projection empowered by cyberinfrastructure. In: CyberGIS for geospatial discovery and innovation. Springer; 2019, p. 171–88. http://dx.doi.org/10.1007/978-94-024-1531-5_9.

[22] Fu Nai Yang, Rios Anne C, Pal Bhupinder, Soetanto Rina, Lun Aaron TL, Liu Kevin, Beck Tamara, Best Sarah A, Vaillant François, Bouillet Philippe, et al. EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival. Nature Cell Biol 2015;17(4):365–75. http://dx.doi.org/10.1038/ncb3117.

[23] Zivanovic Nevena, Öner Deniz, Abraham Yann, McGinley Joseph, Drysdale Simon B, Wildenbeest Joanne G, Crabbe Marjolein, Vanhoof Greet, Thys Kim, Thwaites Ryan S, et al. Single-cell immune profiling reveals markers of emergency myelopoiesis that distinguish severe from mild respiratory syncytial virus disease in infants. Clin Transl Med 2023;13(12):e1507.