BioSkel - Towards a framework for OMICS applications

Valentin Beauvais[1,2], Clémence Couton[3,4,5], Nicolò Tonci[9,10], Sébastien Limet[1,2], Lucile Mollet[3,4,8], Thierry Prazuck[5,6,7], and Sophie Robert[1,2]

[1] Université d'Orléans, INSA CVL, LIFO, UR 4022, Orléans, France
[2] firstname.name@univ-orleans.fr
[3] Centre de Biophysique Moléculaire, CNRS UPR4301, Orléans, France
[4] firstname.name@cnrs-orleans.fr
[5] Centre Hospitalier Universitaire d'Orléans, France
[6] firstname.name@chu-orleans.fr
[7] LI2RSO, Orléans, France
[8] Université d'Orléans, France
[9] Computer Science Department, University of Pisa, Italy
[10] firstname.name@phd.unipi.it

Abstract. The increasing accessibility of next-generation sequencing (NGS) techniques has significantly expanded transcriptomics research. Consequently this leads to major computational challenges as the amount of generated data grows. BioSkel is a framework designed to facilitate the development of bioinformatics workflows for transcriptomics, offering both flexibility and parallelization for shared-memory and distributed-memory architectures. BioSkel allows bioinformaticians to tailor data processing pipelines by integrating custom code while abstracting parallelization complexities. This paper presents new experimental results demonstrating BioSkel applicability to real-world clinical transcriptomic studies.

Keywords: RNA-seq analysis · Transcriptomics application · High-Level Parallel Programming · Distributed Programming

1 Introduction

Through the last decade, cost reductions of new-generation sequencing techniques made nucleic acid-based OMICS experiments more accessible. OMICS refers to studies that identify and quantify entire families of biomolecules (e.g., DNA, RNA, proteins) within biological samples. For example, transcriptomics focus on the full RNA population produced via transcription from DNA. While studying entire transcriptomes has enabled new discoveries, it has also significantly increased the computational demands of processing the resulting large-scale datasets.

RNA-seq is a widely used transcriptomic technique for quantifying transcript (RNA) abundance. To generate meaningful results, large multi-sample datasets

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

across different conditions must be pre-processed. This involves finding the position of short sequences (i.e reads) in a reference genome using an indexed version of the sequence to perform a search. This location determines which features (i.e. genes) are overlapped by the read. A feature counter then tallies how many reads map to each feature across samples. The resulting counts are used for differential expression analysis (DA), comparing conditions regrouping different samples.

These pre-processing tools are often connected into a workflow manager, a high-level language solution such as Nextflow[5] which will handle the execution and exchange between the different tools and scripts. Workflow managers offer modularity, scalability, and reproducibility, but they have two main drawbacks: they often require learning a new language, and most do not support distributed-memory architectures.

In [4], BioSkel is presented as a framework for developing customizable transcriptomic data preprocessing workflows. It provides flexible programming skeletons that let bioinformaticians adjust parameters or integrate custom code tailored to the nature of the data. To handle large datasets, BioSkel also automatically build parallel programs optimized for multicore machines or HPC clusters without the bioinformatician needing to know the technical details.

This paper presents results from the use BioSkel in a real-world transcriptomic study on the effects of cannabidiol (CBD) in long-term suppressed HIV-1 adults. It demonstrates BioSkel's customizability, alignment accuracy comparable to state-of-the-art tools, and scalability on distributed architectures.

2 The BioSkel Framework

In transcriptomics, input data consists of files containing RNA sequences, each file representing a sample (e.g., a patient, a collection date, or a drug treatment). The biological analysis involves comparing how these samples are expressed across genes of a reference genome. This analysis requires data pre-processing, which consists of three computational steps: sequence alignment against a reference, feature counting, and differential expression analysis across samples. This pre-processing is computationally expensive due to the large number of RNA sequences and the high cost of the alignment step.

The BioSkel framework for transcriptomic applications is illustrated Fig. 1. It follows a predefined workflow consisting of three steps: alignment, feature counting, and differential expression analysis. The alignment step is implemented as a programming skeleton. Our approach consists of selecting up to one position per strand in the reference, from which an exact alignment score is computed to assess the relevance of these positions. To determine the best positions, the sequence is divided into small overlapping segments (called seeds) of size n_1, starting from its center, with an overlap of n_2. The alignment positions of each seed are then searched within the reference. This search is optimized using a state-of-the-art library (AWFM)[2], which is reported to be twice as fast for short-read searches compared to the seqan3 library [8]. From all these generated positions, up to three candidates are selected per strand. This selection mainly

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

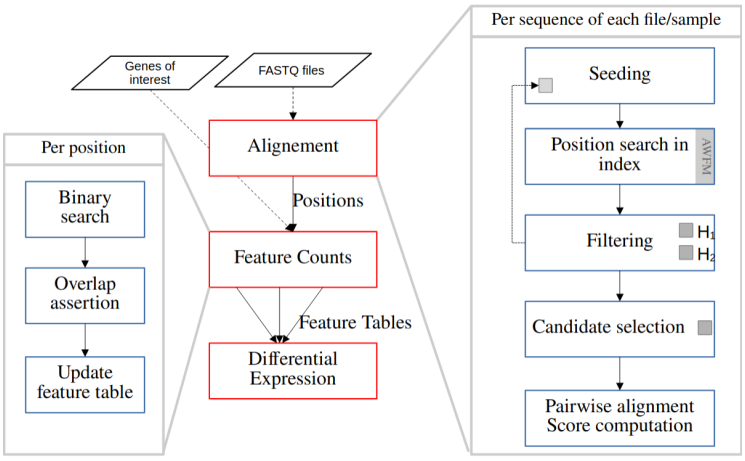

Fig. 1. The predefined workflow of the BioSkel Framework for transcriptomic analysis.

depends on data quality. Our skeleton offers three customizable sections, allowing users to tailor the selection process to their own needs. The first two sections,H_1 and H_2, are heuristics. H_1 is responsible for assessing the quality of a seed based on its set of alignment positions. H_2 evaluates the quality of the entire set of positions generated from the seeds. H_2 estimates the likelihood of finding three suitable candidate positions; if the probability is too low, it can trigger a re-seeding step with optimized n_1 and n_2 values. The last user-configurable section is the selection algorithm, which determines up to two candidate positions used for pairwise comparisons with the reference sequence. Finally, a score is computed based on the exact alignment of the full sequence against the reference at the candidate positions. Among the candidates, we retain the position with the highest score. However, if the best score falls below a predefined threshold T, the alignment is considered unsuccessful.

The feature counting step is fully implemented in our framework. Successful positions are compared to the set of reference features (genes of interest in Fig. 1) to increment the corresponding feature counter. Feature counts are then aggregated for the final differential expression analysis (DA). Bioskel is designed to process bulk RNA barcoding and sequencing datasets (BRB-seq) [1], a 3-end only sequencing method which limits the number of splicing sites. Support for spliced transcripts will be implemented in later version.

DA is a method used to identify genes or transcripts that exhibit significant differences in expression levels between two or more conditions. This analysis typically involves statistical testing to determine which genes are upregulated or downregulated in one condition compared to another. The results provide insights into biological processes, disease mechanisms, and potential therapeutic targets. This step can vary significantly depending on the user's needs. In our

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

framework, it must be implemented by the user. BioSkel facilitates the integration of external codes for this purpose.

To customize the workflow, BioSkel proposes a test mode that runs the computation on a limited number of sequences per file. It also generates logs from various parts of the aligner to help the user fine-tune parameters and adapt code components in the BioSkel skeleton. This avoids running unsuitable configurations on the full dataset and saves time.

The BioSkel framework uses FastFlow [1] to execute its predefined workflows efficiently across both shared- and distributed-memory environments [9]. For full implementation details, we refer the reader to [4].

3 BioSkel on a clinical study

The transcriptomic dataset used for the experiments is a balanced subset of 18 samples of a BRB-seq dataset. This dataset was produced within the scope of a randomized, placebo-controlled, clinical trial assessing the effects of a pharmaceutical grade full-spectrum cannabidiol (CBD) oil at 1 mg/kg twice a day on various biological pathways (i.e. inflammation, autophagy) and gene expression in immune cells of long-term suppressed HIV-1 adults. Trial design and treatment are detailed in [3]. Samples are sorted into three cellular subsets : monocytes (MC), T4 and T8 lymphocytes (T4 and T8) and two time stamps : month 0 (M0) and month 3 (M3). Total RNA extraction from blood samples were sent to a third-party contractor (Alithea Genomics, SA, Épalinges, Switzerland) for library preparation and BRB-seq. The sequencing yielded 2,336,488,125 single-end reads (sequences) of 90 nucleotides, for a total of around 630 Go.

The pre-defined BioSkel workflow customization consists in setting the parameters n_1 and n_2 for seed size, and T as the alignment score threshold. We must also give the implementation of the heuristics H_1 and H_2, along with a seed selection strategy to define three candidates for alignment. Finally, the DA step must be provided.

AWFM index parameters were set as recommended by the library authors [2]. We used $n_1 = 18$ and $n_2 = 6$ (one-third of n_1). Scoring parameters were: match = 0, mismatch = –6, gap open = –5, gap extension = –3, with $T = -60$. Heuristic H_1 flags each seed based on the number of matching positions: empty if none, overflowed if above 10,000. Heuristic H_2 interprets these flags and triggers reseeding when necessary. If fewer than 3 seeds are unflagged, or if most are empty, reseeding occurs with $n_1 - 6$ and $n_2 - 4$. If most are overflowed, reseeding uses $n_1 + 6$ to reduce the number of hits. Candidate selection uses adjacency filtering [10] (see Fig. 2). Among the three seeds with the fewest matches, the one with the fewest positions is chosen as the anchor, the others as comparators. Anchor positions are kept if they have close neighbors within a distance of 200. The anchor with the most neighbors is selected. In case of a tie, the first position in the list is kept. If the anchor has only one position, it is used directly. The DA step compares two user-selected samples (columns in the feature count table). It produces: (1) the raw count table for QC, clustering, and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

accuracy checks; (2) a row-wise ratio between the two columns to help biologists identify features affected by experimental variables (e.g., time, drug). Further analyses (e.g., statistical testing, functional enrichment) are left to tools such as R or Python, as they are study-specific and lightweight computationally, and thus outside the BioSkel scope.

This customization has been done thanks to the test mode available in BioSkel.

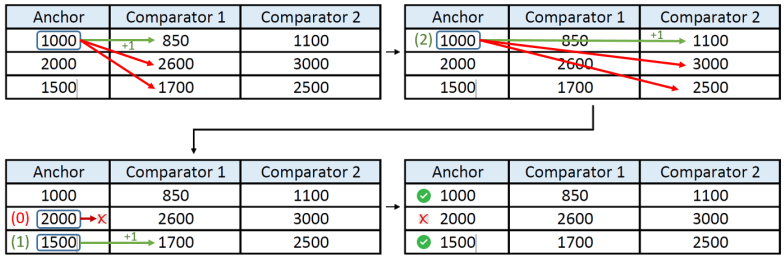

Fig. 2. Principle of our selection algorithm. Comparison tests yield 2 neighboring positions for the position 1000, one for 1000 but none for position 2000. The selection returns 2 candidates : positions 1000 and 1500.

4 Experimental Results

In this section, we assess the performance of BioSkel on a HPC cluster using our custom workflow on subsets of the dataset described in Section 3. Then the ability of BioSkel to produce biologically meaningful results in comparison with data from the contractor and another aligner, Bowtie2 [7], is evaluated.

4.1 Performance evaluation

The experiments were carried out on the LETO cluster hosted by the région Centre-Val-de-Loire. We used between 1 and 4 nodes of LETO which each have 512GB of memory, two AMD Epyc 7702 CPUs running at 2.0GHz for a total of 128 physical cores. Nodes are connected through Infiniband. The workflow and its dependency libraries were compiled with GCC 14.2. All distributed tests presented in this section were executed using the MTCL[6] library and TCP transport protocol of the *FastFlow*library.

We assessed the performance and scalability of our workflow using a hybrid execution model that combines shared-memory and distributed-memory parallelism. Our experiments were conducted on the LETO cluster with varying numbers of nodes N. The size of our clinical study dataset prevented us from conducting performance tests on the entire dataset using the LETO cluster which is a shared resource. We performed four classes of experiments, processing subsets of the dataset containing 1, 2, 4, and 8 files, respectively. The files were chosen

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

to have different sizes to test load balancing, while also ensuring sequence diversity, as the alignment complexity highly depends on the data (i.e., the number of seeds and the number of positions per seed). The files correspond to different participants, sampling times, and cell types.

Our results demonstrate good scalability across different configurations. However, we noticed that scalability was more favorable in the 4-file experiment compared to other cases. It illustrates how workload distribution efficiency is influenced by dataset characteristics. To address this, the module in charge of reading the data on disk distributes chunks of each file of the dataset using a round-robin method. On the full dataset, this approach should ensure a well-balanced workload distribution.

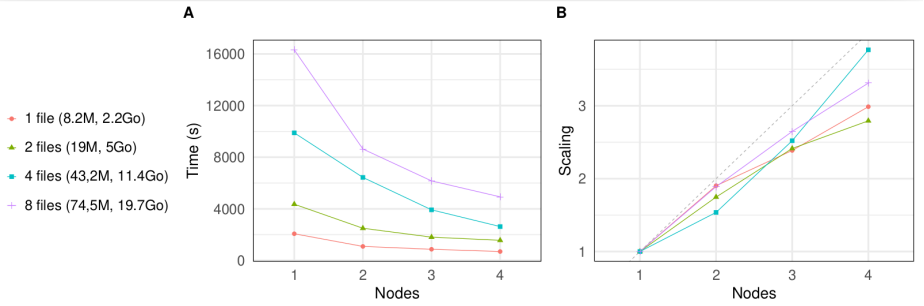

Fig. 3. Distributed-memory performance evaluation of BioSkel on the *LETO* cluster. (A) illustrates the execution times and (B) the scalability traces for each dataset size according to the number of nodes. Ideal scaling is shown by a grey dashed line. Shown results are averages of 6 run times per conditions. The total number of processed sequences are in millions and the cumulative file size in Gigabytes.

4.2 Accuracy monitoring

In [4], we compared the accuracy of a basic version of our aligner against Bowtie2 on synthetic dataset and reached comparable results. Here, we monitored the accuracy of both solutions against a biological dataset. We used the count tables (feature counter output), which contain informations on how each feature (i.e. transcript) is represented in each sample. We performed the same tests using UMI-deduplicated, pre-processed count tables provided by the contractor.

The accuracy test labeled Bowtie2 was executed by aligning with Bowtie2 version 2.4.4. Alignment results were then mapped to featureCounts version 2.0.3 from the Subread package. Accuracy tests were performed with default parameters for Bowtie2 and FeaturesCounts except for the -g option, which was set to select "gene" coordinates. Both aligners used the same parameter limits of -60 for the scoring. The correlation between samples was computed with R version 4.4.2. and clusterization was done with the complexHeatmap package. All solutions were aligned against the homo sapiens hg38 human genome.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

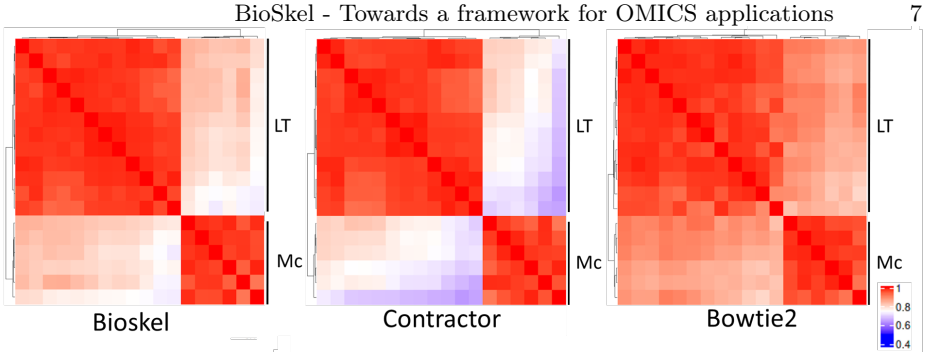

Fig. 4. Evaluation of biological relevance of BioSkel output. Each heatmap shows the clusterization of inter-sample correlation computed using the count tables from BioSkel, Bowtie2 or provided by the contractor. The first heatmap was generated with specific cell-marker counts using BioSkel to control the quality of the dataset. The two others were realized with the whole count tables with 2 different aligners.

We confirmed the quality of the BRB-seq data by regrouping samples using clusterization by inter-sample correlation with specific cell marker counts. We used the following markers for the clusterization : CD3G, CD3D, CD3E, CD8A, CD8B, PRF1, GZMB, GZMA, CD4, CD14, PTPRC, FCGR3A, FCGR3B, CD16, NCAM1. We obtained a clean separation between all cell types (monocyte, T4 and T8) indicating good data quality, with all three solutions.Unlike synthetic data, there is no reliable technique to obtain the absolute solution of a biological dataset. Therefore, the accuracy of BioSkel was measured with a data-oriented method. Since cell populations have different global expression profiles (i.e. count table), clusterization by inter-sample correlation using these profiles tends to regroup similar cell populations. We applied this clusterization to the profiles obtained using BioSkel, Bowtie2 or provided by the sequencing company. As shown in Fig. 4, all three methods produced good clustering with minimal variation, separating monocytes and lymphocytes (T4 and T8). The small variations between the results are likely due to differences in the internal design of the aligners.

5 Conclusion

This paper presents the framework BioSkel and how it can be configured to process a real case transcriptomic study. We presented experimental results that assess both the accuracy of the framework and its scalability. These results demonstrate the potential of the framework for processing large volumes of data on distributed-memory architectures. They also show the relevance of BioSkel to design workflows in the OMICS context.

In the context of this clinical study, we plan to run the Bioskel workflow on the whole dataset using a national computing center to confirm the results presented in this article. In addition these experiments will help us to develop better heuristics and search methods to improve speed without compromising

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30

accuracy. Looking ahead, we seek to make BioSkel even more flexible by introducing additional customization points within the pre-defined workflow.

Acknowledgments

This work was funded by the APR-IA project BioSkel and APR-IR CannApp, supported by the Région Centre Val de Loire. C. Couton received a fellowship from ANRT (Association Nationale Recherche Technologie) and ANRS-MIE (Agence Nationale Recherche SIDA). The authors thank participants and hospital staff for their involvement in the clinical trial analysed herein. The authors benefited from the use of the cluster at the Centre de Calcul Scientifique en région Centre-Val de Loire.

References

1. Alpern, D., Gardeux, V., Russeil, J., Mangeat, B., Meireles-Filho, A.C.A., Breysse, R., Hacker, D., Deplancke, B.: Brb-seq: ultra-affordable high-throughput transcriptomics enabled by bulk rna barcoding and sequencing. Genome Biology **20**(1), 71 (2019). https://doi.org/10.1186/s13059-019-1671-x
2. Anderson, T., Wheeler, T.J.: An optimized fm-index library for nucleotide and amino acid search. Algorithms for Molecular Biology **16**(1), 25 (2021). https://doi.org/10.1186/s13015-021-00204-6
3. Barré, T., Couton, C., Mourad, A., Carrieri, P., Protopopescu, C., Klein, H., de Dieuleveult, B., Hocqueloux, L., Mollet, L., Prazuck, T.: Limited impact of cannabidiol on health-related quality of life of people with long-term controlled hiv: A double-blind, randomized, controlled trial. Open Forum Infectious Diseases **11**(9), ofae492 (08 2024). https://doi.org/10.1093/ofid/ofae492
4. Beauvais, V., Tonci, N., Robert, S., Limet, S.: Parallelizing RNA-seq analysis with BioSkel: A FastFlow based prototype. International Journal of Parallel Programming **53**(2) (2025). https://doi.org/10.1007/s10766-025-00786-3
5. Di Tommaso, P., et al.: Nextflow enables reproducible computational workflows. Nature Biotechnology **35**(4), 316–319 (2017). https://doi.org/10.1038/nbt.3820
6. Finocchio, F., Tonci, N., Torquati, M.: Mtcl: a multi-transport communication library. In: European Conference on Parallel Processing. pp. 55–67. Springer (2023). https://doi.org/10.1007/978-3-031-50684-0_5
7. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with bowtie 2. Nature Methods **9**(4), 357–359 (2012). https://doi.org/10.1038/nmeth.1923
8. Reinert, K., Dadi, T.H., et al.: The seqan c++ template library for efficient sequence analysis: A resource for programmers. Journal of Biotechnology **261**, 157–168 (2017). https://doi.org/10.1016/j.jbiotec.2017.07.017
9. Tonci, N., Torquati, M., Mencagli, G., Danelutto, M.: Distributed-memory fastflow building blocks. International Journal of Parallel Programming **51**(1), 1–21 (2023). https://doi.org/10.1007/s10766-022-00750-5
10. Xin, H., Lee, D., Hormozdiari, F., et al.: Accelerating read mapping with fasthash. BMC Genomics **14**(Suppl 1), 1–13 (2013). https://doi.org/10.1186/1471-2164-14-s1-s13

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_30