Accelerating Cloud-Based Transcriptomics: Performance Analysis and Optimization of the STAR Aligner Workflow

Piotr Kica[1,2], Sabina Lichołai[1,3], Michał Orzechowski[1,2,3], and Maciej Malawski[1,2]
{p.kica, s.licholai, m.orzechowski, m.malawski}@sanoscience.org

[1] Sano Centre for Computational Medicine, Kraków, Poland
[2] AGH University of Kraków, Faculty of Computer Science, Poland
[3] Academic Computer Centre Cyfronet AGH, Kraków, Poland

Abstract. In this work, we explore the Transcriptomics Atlas pipeline adapted for cost-efficient and high-throughput computing in the cloud. We propose a scalable, cloud-native architecture designed for running a resource-intensive aligner – STAR – and processing hundreds of terabytes of RNA-sequencing data. We implement optimization techniques that significantly reduce cost and execution time. The impact of particular optimizations is measured in medium-scale experiments followed by a large-scale experiment that leverages all of them and validates the design. Early stopping optimization allows us to reduce the total alignment time by 23%. For the cloud environment, we identify suitable EC2 instance types and verify the applicability of spot instances usage.

Keywords: Transcriptomics · Optimization · STAR · Alignment · High-Throughput Computing · Cloud-computing · AWS

1 Introduction

The cloud is, in many cases, the infrastructure of choice for large-scale genomic pipelines, as it promises scalability, cost efficiency, and availability of on-demand resources. There are multiple recent examples of pipelines that have been developed for genomic data, built using cloud services [6,13,15]. Such cloud-native pipelines take advantage of the capabilities of cloud services, allowing parallelism, scalability, and elasticity, often with autoscaling, and follow the principles of infrastructure as code for application deployment and environment setup. However, cloud infrastructure is complex, as there are multiple services offered with a wide range of configuration options, so exploiting the full potential of clouds requires combining application-specific expertise with the ability to fine-tune cloud infrastructure configuration. For example, there are multiple compute service options, storage systems, databases, and queuing/messaging systems, etc., so making the right choice becomes a challenge. In this paper, we present a case

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

study of running a Transcriptomics Atlas pipeline in the AWS cloud. It is a data- and compute-intensive pipeline, based on a sequence aligner – STAR [9] – that processes hundreds of terabytes of RNA-seq data. Our aim is to answer research questions, such as: (1) How to take advantage of the intermediate results to reduce time and cost? (2) How to select the optimal level of parallelism within a single node? (3) Which instance types are the most cost-efficient for alignment? (4) How suitable are spot instances for running resource-intensive aligners?

2 Background and related work

Transcriptome and NGS sequencing. The human transcriptome consists of different types of RNA, ranging from messenger RNA (mRNA) encoding proteins to various forms of non-coding RNA with mostly regulatory functions [10]. Transcriptome-releated analyses are essential in modern medical research and typically performed in a comparative context using next-generation sequencing (NGS) data. Lack of comprehensive NGS data under standard conditions significantly increases the cost of experiments [7] and in response to this, we designed the Transcriptomics Atlas, where data from a representative collection of human tissues were processed in a uniform manner. NGS is based on short reads and enables rapid and precise identification of the transcriptome composition within a biological sample. However, due to the limited length of reads, subsequent assembly into a complete transcriptome necessitates specialized bioinformatics algorithms in a process called alignment [3].

RNA-sequencing in the cloud. In [17], authors explore different parallel computing strategies and limitations for multiple genomic tools, including architecture-aware and data-storage optimizations. Research on STAR-based workflow as well as cost and throughput analysis for cloud and HPC experiments are carried out in [16]. Pseudoaligners (e.g. Salmon, Kallisto) are recommended by [13] when cost plays a critical role. Research carried out in [8] shows that serverless computing for RNA sequencing is a valid approach when high parallelism is the end goal, with HiSat2 running on AWS Lambda. However, deploying STAR to serverless services is more challenging compared to less resource-intensive aligners. Although possible, it is not recommended for large-scale processing due to decreased cost-efficiency compared to VM-based solutions [12]. Furthermore, in [4] authors moved an HPC workflow to serverless services and identified multiple challenges with efficient data partitioning, transfer and insufficient object storage performance. Our previous work regarding the Transcriptomics Atlas [5] focused on understanding the pipeline requirements for a similar workflow in HPC and the cloud.

3 Pipeline and cloud architecture

Pipeline description. As presented in Fig. 1, the first phase consists of accessing an *SRA* file using `prefetch` and converting it into *FASTQ* with `fasterq-dump`. The next and most important, time-consuming step is the alignment with `STAR`.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

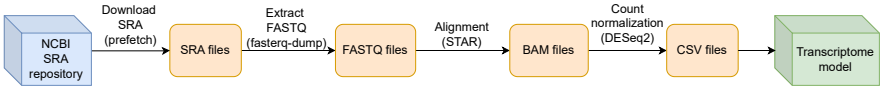

Fig. 1: Transcriptomics Atlas Pipeline.

Finally, the acquired *BAM* file is normalized using `DESeq2`. Instead of STAR, one can use alternative aligners (HiSat2) or pseudoaligners (Salmon). We are running STAR v2.7.10b with "`-quantMode GeneCounts`" option. The steps are connected using a Python script, and the current implementation of the Transcriptomics Atlas project is publicly available on GitHub under the MIT license [2].

Input dataset. Data were obtained from the NCBI Sequence Reads Archive (SRA) [1], focusing on nucleotide sequences from human samples, filtered by tissue type. The query for the SRA database targeted publicly available, human-origin data sequenced with Illumina machines. We downloaded metadata for matching SRA IDs and selected those with compressed sequence sizes between 200MB and 30GB. We define the range by taking into account the size of libraries for typical transcriptome sequencing and output from the most commonly used wet lab sequencing protocols. For the Transcriptomics Atlas, we aim for 100–200 (with a good mapping rate) per tissue, selecting up to 400 samples per tissue, resulting in 7216 files totalling 17TB of SRA data.

Resource requirements. STAR is a resource-intensive aligner that uses a precomputed genome index in the alignment step, which has to be loaded into the system memory. The generation of such an index is a one-time task. Depending on the type and release version of the genome, the index differs in size. We use the human genome of the "Toplevel" type distributed by Ensembl [14]. Previous work [11] showed that an older release (108) results in an index of 85GiB in size, and using a newer one (release 111) is much smaller (29.5GiB) and faster (12 times). STAR requires additional memory for sorting a BAM file, usually 1-2 GiB, but outliers may require even 20.5 GiB. The pipeline requires enough disk space to handle intermediate files such as *FASTQ* files along with *SRA* and *BAM* files. The fasterq-dump tool creates *FASTQ* files, on average, 7.5 times bigger than the original *SRA* file; however, outliers can be even 17 times larger. Moreover, additional space is required during conversion. This use case focuses on files within the 200MB - 30GB range. Therefore, we can estimate that the required space should not exceed 550GiB.

Cloud architecture for the Transcriptomics Atlas pipeline is presented in Fig. 2. The main processing is performed on a virtual cluster of EC2 spot instances launched from a custom machine image containing all the required software. The initial step for each worker is to connect to an NFS instance and load the STAR index into memory. Subsequently, workers acquire the SRA IDs from the queue and process them using the pipeline (Fig. 1). The transcriptomics results are stored in a dedicated S3 bucket. Execution metadata are gathered for performance analysis and saved in a Dynamodb table. Metrics are saved using the CloudWatch service, which gives insight into resource utilization. The pro-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

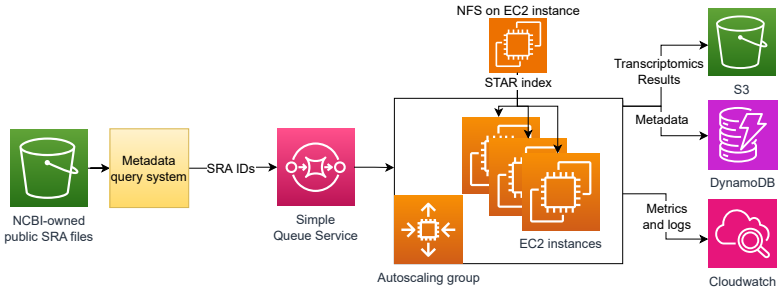

Fig. 2: Cloud architecture for Transcriptomics Atlas Pipeline.

posed approach is easily scalable and adaptable for similar workflows. Having extensive control over the underlying compute resources allows us to fine-tune the configuration for given requirements, which improves cost-efficiency.

4 Application-specific Optimizations

Early stopping during alignment. Early stopping is a common method to stop the training of machine learning models when the desired accuracy is achieved. We apply a similar approach to alignment by discarding low-quality or invalid sequences during processing. It is possible, as STAR reports on the intermediate mapping rate at runtime, which allows the identification of such sequences. As shown in [11], utilization of live metrics from *Log.progress.out* file can boost alignment throughput by up to 19.5%. This feature is beneficial when we cannot determine the quality of the FASTQ beforehand.

For the Transcriptomics Atlas project, the mapping rate threshold is set at 30%. However, this is highly dependent on the use case. Pipelines that utilize STAR in a similar scenario and require sequences of the highest quality will greatly benefit from this feature. In Fig. 3, we present an extended analysis based on the experiment carried out in [11]. If we set the threshold at 80%, we would reduce the total compute time by 60%. The minimal number of processed spots was set to 10%.

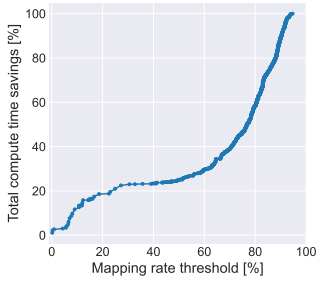

Fig. 3: Impact of early stopping threshold on total alignment time.

Cost-efficient allocation of cores. The recommended approach for STAR aligner is to match the thread count with the number of available cores in the node. Authors of the original STAR publication [9] claim that "STAR exhibits close to linear scaling of the throughput rate with the number of threads". However, the original research lacks detailed performance analysis, which is important to maximize CPU efficiency. We decided to test the scalability of STAR

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

to confirm these claims and find an optimal number of cores per node, which will improve the throughput of the pipeline. The test suite consists of 3 *FASTQ* files of different sizes. Execution times are measured on two different 16-vCPU instance types - with and without Simultaneous Multi-Threading (SMT).

The test results are presented in 4, and we see the benefit of the increased number of threads. However, there is a noticeable drop in efficiency - for 16 threads and m7a.large instance, we get 84% and 72% efficiency for 16GiB file and 81GiB respectively. This is especially visible for the m6a.large instance, which uses SMT and exceeding 8 threads further decreases efficiency. Based on the acquired metrics, we focused on 8-vCPU instances for the best cost-efficiency.

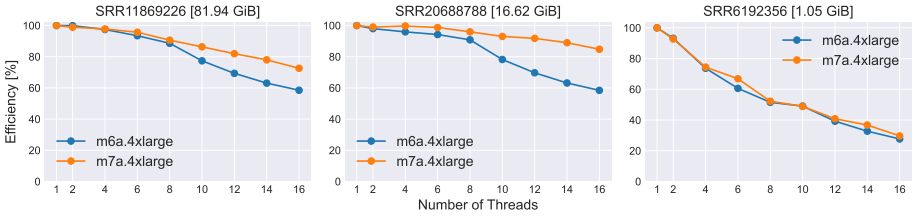

Fig. 4: Efficiency of the STAR aligner.

5 Cloud Infrastructure Optimizations

Instance type comparison in AWS. According to the requirements in Section 3 and knowing that STAR is a memory-intensive program, we decided to focus on instances with a higher memory per CPU factor with at least 64GiB of RAM. Using instance types which have more cores but less memory per CPU may require faster block storage and result in increased under-utilization during other, much less CPU-intensive steps (e.g. *prefetch*). Using more cores in a single worker node would also reduce CPU efficiency during alignment as described in Section 4. The selected instance types of the current generation that meet these requirements are compared in Table 1. This table also presents the total cost and time for performing STAR alignment on 50 random *FASTQ* files. The results indicate r7a.2xlarge as the fastest and cheapest type. However, when using spot instances, the availability of a given type is also important.

Table 1: Cost-efficiency analysis of selected instance types.

Instance type	vCPU	Cores	RAM [GiB]	On-demand price [h]	Total STAR execution time [h]	Total cost
r6a.2xlarge	8	4	64	$0.4536	8.00	3.63 $
r6i.2xlarge	8	4	64	$0.5040	8.04	4.05 $
r7a.2xlarge	8	8	64	$0.6086	5.48	3.33 $
r7i.2xlarge	8	4	64	$0.5292	7.66	4.05 $

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

Cost-efficiency of spot instances. Spot instances on AWS offer compute resources at a lower cost (depending on instance type and current market demand). However, such instances can be terminated with a 2-minute notice. For example, an r7a.2xlarge instance can be acquired with 50%-60% discount. Our use case fits this model as the optimized pipeline runs relatively quickly (mean = 8 min). Unfortunately, interruptions result in an additional STAR initialization phase and restarting the computations on a new instance. However, with a good configuration (instance types with a low interruption rate), using spot instances should result in relatively stable computations. In Fig. 5. we present the processing 1000 *SRA* files on r7a.2xlarge instances. During the experiment, only five interruptions occurred and resulted in a loss of <1% of the total running time.

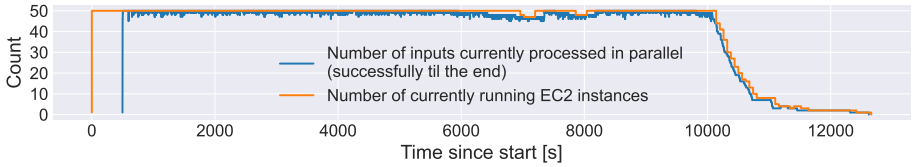

Fig. 5: Spot instances usage experiment timeline.

6 Large-scale Experiment

The goal is to test the pipeline on a larger scale, measure, and analyze resource usage. We use the input data from Section 3 and the configuration:

- EC2 (Spot): 50 r7a.2xlarge instances (EBS: 550GB, GP3, 500MiB/s, 3000IOPS)
- Input: 7216 *SRA* files (2.5GB avg, 17.9TB total size, max=29.9GB)
- Index: Based on Toplevel human genome, release 111, 29.5GB size.

The experiment used 1102.5 node hours in total and the implemented optimizations gave the expected improvements. The timeline is presented in Fig. 6. We processed 130TB of *FASTQ* data and acquired an average mapping rate between 57%-87%, depending on the tissue. Early stopping feature reduced the total run time of STAR by about 23%. Using spot instances saved 50% the compute costs, but 138 interruptions occurred, wasting only 2.9% of the total instances' run time. The average CPU utilization across all instances was about 58% for the entire pipeline and 78% for STAR exclusively. 93% of all instances run time the RAM utilization was between 45% and 55%, and only 1.7% required more than 60% of the instance memory, suggesting an area for improvement. STAR accounted for 71% of the total workload time. In Fig. 7, we show aggregated CPU and memory usage during STAR for normalized metrics gathered during alignments longer than 10 minutes (n=1091). The estimated cost of the experiment is about $477 - including compute (70%), storage (18.5%) and data transfer costs (11.5%). This is equivalent to about $0.066 per *FASTQ* file.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

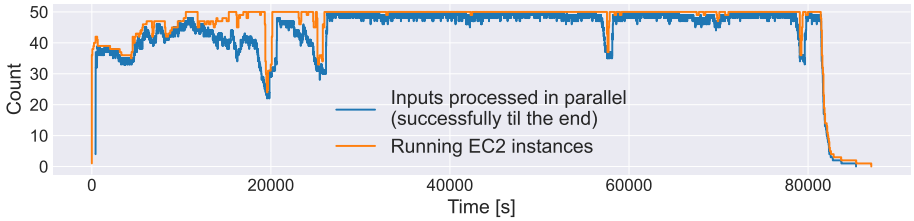

Fig. 6: Large experiment timeline for instances and successfully computed files.

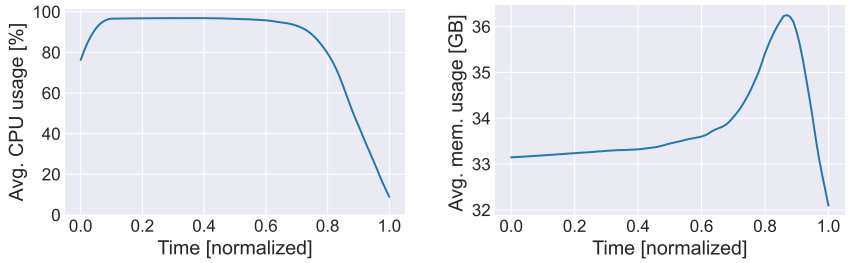

Fig. 7: STAR aggregated and normalized metrics for CPU and memory usage.

7 Conclusions

This work presents the cloud architecture for the Transcriptomics Atlas pipeline with the STAR aligner as its core. The optimizations described here significantly increased performance and throughput. Analysis of a large-scale experiment showed that early stopping saved 23% of the total running time of STAR. For use cases with a high mapping rate threshold, this feature would be even more beneficial. In addition, we observe that the pipeline is a great fit for spot instances and reduces computational costs by 50%-60%. Analysis of STAR's efficiency will help to choose the right configuration in order to maximize the throughput in similar scenarios. We identified one of the best instance types on AWS for alignment in terms of processing time and cost. Many insights in this work are applicable outside the cloud environment, extending the research results for HPC centres and workstations. As concluded in [11], a faster STAR alignment can also improve the time required for a clinician to make a diagnosis.

Acknowledgments. The publication is supported by the Polish Minister of Science and Higher Education, contract number MEiN/2023/DIR/3796; EU Horizon 2020 Teaming grant agreement No 857533; IRAP program of the Foundation for Polish Science MAB PLUS 2019/13; and EU Horizon Europe grant NEARDATA No 101092644.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31

References

1. NCBI NIH: The Sequence Read Archive (2024), https://www.ncbi.nlm.nih.gov/sra
2. Transcriptomics Atlas (2024), https://github.com/SanoScience/NearData
3. Alser, M., Rotman, J., Deshpande, D., Taraszka, K., Shi, H., Baykal, P.I., Yang, H.T., Xue, V., Knyazev, S., Singer, B.D., et al.: Technology dictates algorithms: recent developments in read alignment. Genome biology **22**(1), 249 (2021)
4. Arjona, A., Gabriel-Atienza, A., Lanuza-Orna, S., Roca-Canals, X., Bourramouss, A., Chafin, T.K., Marcello, L., Ribeca, P., García-López, P.: Scaling a Variant Calling Genomics Pipeline with FaaS. In: Proceedings of the 9th International Workshop on Serverless Computing. pp. 59–64 (2023)
5. Bader, J., Belak, J., Bement, M., Berry, M., Carson, R., Cassol, D., Chan, S., Coleman, J., Day, K., Duque, A., et al.: Novel approaches toward scalable composable workflows in hyper-heterogeneous computing environments. In: Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis. pp. 2097–2108 (2023)
6. Camacho, C., Boratyn, G.M., Joukov, V., Vera Alvarez, R., Madden, T.L.: ElasticBLAST: accelerating sequence search via cloud computing. BMC bioinformatics **24**(1), 117 (2023). https://doi.org/10.1186/s12859-023-05245-9
7. Casamassimi, A., Federico, A., Rienzo, M., Esposito, S., Ciccodicola, A.: Transcriptome profiling in human diseases: new advances and perspectives. International journal of molecular sciences **18**(8), 1652 (2017)
8. Cinaglia, P., Vázquez-Poletti, J.L., Cannataro, M.: Massive parallel alignment of rna-seq reads in serverless computing. Big Data and Cognitive Computing **7**(2), 98 (2023). https://doi.org/10.3390/bdcc7020098
9. Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., Gingeras, T.R.: STAR: ultrafast universal RNA-seq aligner. Bioinformatics **29**(1), 15–21 (2013). https://doi.org/10.1093/bioinformatics/bts635
10. Gibbs, R.A.: The human genome project changed everything. Nature Reviews Genetics **21**(10), 575–576 (2020). https://doi.org/10.1038/s41576-020-0275-3
11. Kica, P., Lichołai, S., Orzechowski, M., Malawski, M.: Optimizing Star Aligner for High Throughput Computing in the Cloud. In: 2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops). pp. 162–163. IEEE (2024). https://doi.org/10.1109/CLUSTERWorkshops61563.2024.00039
12. Kica, P., Orzechowski, M., Malawski, M.: Serverless approach to running resource-intensive star aligner. arXiv preprint arXiv:2504.05078 (2025)
13. Lachmann, A., Clarke, D.J., Torre, D., Xie, Z., Ma'ayan, A.: Interoperable RNA-Seq analysis in the cloud. Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms **1863**(6), 194521 (2020)
14. Martin, F.J., Amode, M.R., Aneja, A., Austine-Orimoloye, O., Azov, A.G., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., et al.: Ensembl 2023. Nucleic acids research **51**(D1), D933–D941 (2023). https://doi.org/10.1093/nar/gkac958
15. Wiewiórka, M., Szmurło, A., Stankiewicz, P., Gambin, T.: Cloud-native distributed genomic pileup operations. Bioinformatics **39**(1), btac804 (2023)
16. Wilks, C., Zheng, S.C., Chen, F.Y., Charles, R., Solomon, B., Ling, J.P., Imada, E.L., Zhang, D., Joseph, L., Leek, J.T., et al.: recount3: summaries and queries for large-scale RNA-seq expression and splicing. Genome biology **22**, 1–40 (2021)
17. Zou, Y., Zhu, Y., Li, Y., Wu, F.X., Wang, J.: Parallel computing for genome sequence processing. Briefings in Bioinformatics **22**(5), bbab070 (2021)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_31