PAPER

CellHeap: A scRNA-seq workflow for large-scale bioinformatics data analysis

Maria Clicia S. Castro,[4] Vanessa S. Silva,[1] Maiana O. C. Costa,[3] Helena S. I. L. Silva,[5] Maria Emilia M. T. Walter,[5] Alba C. M. A. Melo,[5] Kary Ocaña,[3] Marcelo T. dos Santos,[3] Marisa F. Nicolas,[3] Anna Cristina C. Carvalho,[2] Andrea Henriques-Pons[2] and Fabrício A. B. Silva[1,*]

[1] Scientific Computing Program,Oswaldo Cruz Foundation, Av. Brasil, 4365 - Manguinhos, 21040-900, Rio de Janeiro, Brazil, [2] Laboratory of Innovations in Therapies, Education and Bioproducts,Oswaldo Cruz Foundation, Av. Brasil, 4365 - Manguinhos, 21040-900, Rio de Janeiro, Brazil, [3]Computational Modeling Department, National Laboratory for Scientific Computing, Av. Getúlio Vargas, 333, Petrópolis, 15651-075, Rio de Janeiro, Brazil, [4]Department of Informatics and Computer Science, State University of Rio de Janeiro, Av. São Francisco Xavier, 524, Maracanã, 20550-900, Rio de Janeiro, Brazil and [5]Department of Computer Science, University of Brasilia, Campus Universitário Darcy Ribeiro, Asa Norte, Brasilia, 71670-260, Federal District, Brazil

*Corresponding author. fabricio.silva@fiocruz.br

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Several hundred terabytes of single-cell RNA-seq (scRNA-seq) data are available in public repositories. These data refer to various research projects, from microbial population cells to multiple tissues, involving patients with a myriad of diseases and comorbidities. An increase to several Petabytes of these scRNA-seq data is a realistic prediction for coming years. Therefore, thoughtful analysis of these data requires large-scale computing infrastructures and software systems optimized for such platforms to generate correct and reliable biological knowledge. This paper presents CellHeap, a flexible, portable, and robust platform for analyzing large scRNA-seq datasets, with quality control throughout the execution steps, and deployable on platforms that support large-scale data, such as supercomputers or clouds. In addition, CellHeap includes an efficient solution to make use of parallel computational resources, improving total execution time. Furthermore, we show a case study of a this platform, with its deployment in the Brazilian Santos Dumont supercomputer, which was used to process dozens of Terabytes of COVID-19 scRNA-seq raw data. Our results show that most of the total execution time is spent in CellHeap initial phases and that there is great potential for a parallel solution to speed up scRNA-seq data analysis significantly. Additionally, we analysed the produced results, with a focus on particular modulations of Fc receptors considering mild and severe cases verified to be coherent from a biological perspective.

Key words: single-cell RNA-seq, bioinformatics workflow, high-performance computing, COVID-19

Introduction

In recent years, single-cell RNA sequencing (scRNA-seq) technologies, have been increasingly used to characterize cellular and molecular heterogeneity by profiling transcriptomes at the single-cell level [1]. These technologies have produced many publicly available single-cell RNA-seq (scRNA-seq) datasets. On the other hand, many sophisticated computational tools have been developed to generate knowledge of cellular processes and pathways by processing those large volumes of data. Together, these data and tools have enabled studies of cell heterogeneity and developmental dynamics across diverse biological systems, which allows a better understanding of the underlying biological processes in organisms [2, 3] and complex diseases [1, 4, 5, 6].

Specifically, scientific workflows (or pipelines) are the primary tool used to perform computational modeling based on scRNA-seq data [7], which may involve different sequencing protocols [3, 8, 9]. However, there has yet to be a generic workflow for scRNA-seq which require mastering complex computational techniques, such as high-performance computing implementation and sophisticated analytic data science algorithms. Currently, users must develop scripts not guaranteed to cover the requirements of executing large and complex scRNA-seq experiments. Given this context, in this paper we propose a generic bio-computation service platform that focuses on high-performance computing platform deployments.

Scientific workflows [10] allow for better modeling and execution of tasks and dataflows to solve problems in diverse areas of research. To create a workflow, scientists often use an *ad hoc* strategy, writing from scratch a set of scripts that connect tasks. This approach leads to different workflows, each targeting a specific situation, usually creating slightly different variations of the same idea. Consequently, reproducibility, reliability, and scalability in scientific experiments may be at stake [11].

ScRNA-seq offer considerable promise in dissecting the heterogeneity of immune responses among individuals and exposing the molecular mechanisms of diseases. In particular, these technologies have been used to study coronavirus disease (COVID-19) pathogenesis [4, 12, 13, 14, 15, 16, 17, 18, 19].

© The Author 2023.

COVID-19 is an ongoing global infectious disease caused by the SARS-CoV-2 virus. ScRNA-seq have bostered the understanding of each phase of this disease and other aspects, such as differential gene expression in the single-cell multi-omics analysis of the immune response in COVID-19 [20]. It is noteworthy that most of the workflows used in the scRNA-seq studies of COVID-19 are not flexible, are built in an *ad hoc* manner, and run in a standalone desktop [14, 15, 16, 17] or on a supercomputer [8, 18].

We claim that when sophisticated bioinformatics tools are created and updated regularly, scientific workflows must be flexible, enabling users to choose between tools that execute the same task. In addition, workflows should be extensible, allowing new modules or phases to be integrated. Moreover, a workflow should also be portable, executable on a standalone platform, such as a desktop server, or on a high-performance computing platform, such as a supercomputer. Another critical aspect is that a workflow for scRNA-seq studies usually analyses samples of hundreds of Gigabytes, requiring execution on a high-performance computing platform.

We present CellHeap, a flexible, extensible, and portable platform for scRNA-seq analysis, which processes hundreds of Gigabytes of scRNA-seq raw data on supercomputer environments. The CellHeap platform is composed of five phases. In each one, CellHeap includes many computational tools and couples them such that inputs and outputs consume/generate data in a flow that meets requirements for subsequent phases of analysis. CellHeap supports the construction of different workflows according to the specific goals of each research project and the associated biological question, as discussed in the following sections. Also, many datasets are available in public repositories, and CellHeap supports several input data formats and sequencing platforms.

We deployed the CellHeap platform in a case study with COVID-19 data on the Santos Dumont (SDumont) supercomputer. Initially, we selected and curated publicly available COVID-19 scRNA-seq raw datasets originally processed in a sequential workflow. Then, based on the sequential workflow execution, we analysed data storage, studied computational performance, and proposed a strategy to improve execution time. In more detail, using scRNA-seq datasets of COVID-19 [4], we parallelized the workflow's initial phases, which are the most time-consuming. Results such as the reduction in the execution time of CellHeap's first stages for the samples representing the severe COVID-19 cases from five days (sequential) to one day and one hour (parallel) demonstrate that a simple parallel solution has great potential to speed up the workflow.

This paper is organized as follows: Section 2 introduces the CellHeap platform. Section 3 describes our case study, which deployed CellHeap in the Santos Dumont supercomputer (SDumont) considering scRNA-seq datasets of COVID-19. Section 4 presents the results obtained from the workflow using COVID-19 datasets, discusses the time and space storage needed to execute the workflow on a high-performance platform, and the results obtained are investigated from a biological perspective. Finally, Section 5 concludes this paper.

The CellHeap platform

This section describes the flexible, extensible, portable, and robust CellHeap platform for scRNA-seq customized bioinformatics workflows. CellHeap includes quality control to ensure correct results and relies on high-performance computational support for parallelizing and distributing computational tasks.

CellHeap comprises five phases (Figure 1): sample curation; data pre-processing; quality control; dimensionality reduction and clustering analysis; and advanced cell level and gene level analysis. An optional phase, sample aggregation, can also be deployed into the CellHeap workflow. The CellHeap source code is available at https://github.com/FioSysBio/CellHeapRelease.

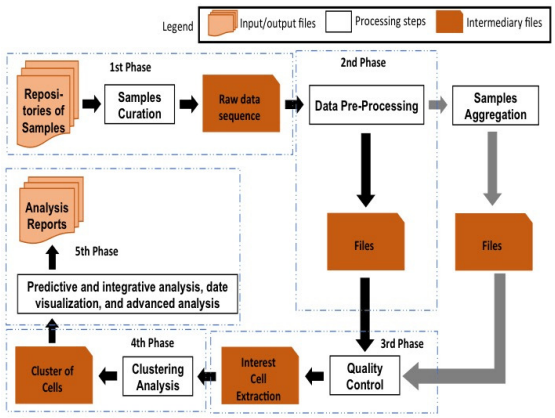

Fig. 1. CellHeap platform conceptual view. Dashed rectangles identify the five phases of the workflow.

The CellHeap phases

The CellHeap platform is fully configurable to the scientific question under investigation. In this section, we describe the phases of the CellHeap platform, along with softwares widely used in recent years.

Phase 1 - Sample curation

Public repositories, such as the Gene Expression Omnibus (GEO), contain specific samples from a single cell (e.g., tissue sample). Applying a pre-filter protocol to the scRNA-seq samples, based on several inclusion/exclusion criteria, indicates their suitability. In CellHeap, we used the same protocol as the Expression Atlas of the European Molecular Biology Laboratory (EMBL)[1].

CellHeap only includes samples that fulfill all criteria, which was guaranteed using NCBI Sequence Read Archive (SRA) toolkit tools to download and validate datasets. The `prefetch` tool downloads the selected raw sequence data in *.sra* format, and samples can be identified by the *SRR prefix + ID from GEO*. The `vdb-validate` tool authenticates the successfully downloaded data. Finally, samples are converted into *.fastq* format files using the `fasterq-dump` tool according to the required input in the next phase.

Phase 2 - Data pre-processing

This phase creates a gene count matrix from the raw sequencing data. `Cellranger`[2], a package of 10x Genomics Chromium [21],

[1] https://www.ebi.ac.uk/gxa/sc/help.html

[2] https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger

is widely used for read mapping, cell demultiplexing, and to generate the cell-wise unique molecular identifier (UMI)-count table. The `cellranger count` software is designed to align reads with a reference genome. It receives the compressed *.fastq* input files and a reference genome to output the barcode matrix feature.

Phase 3 - Quality control

The quality control (QC) phase removes artifacts to ensure that the processed cells are single and not experimentally damaged.

Data quality control for filtering sequenced cells can be performed using `Seurat` [22] and `SoupX` [23] packages. `Seurat` aims to enable users to identify and interpret sources of heterogeneity from single-cell transcriptomic measurements and to integrate diverse types of single-cell data. `SoupX` [23] estimates a fraction of contamination (*rho*) in the samples (reads) through the *autoEstCont* function.

Phase 4 - Clustering analysis

This phase contains the following steps:

- Data integration [24, 25];
- Dimensionality reduction of scRNA-Seq data [26, 27];
- Clustering of cell-level in scRNA-Seq [28];
- Cell-type annotation by identifying cell groups based on gene expression profiles.

Seurat's Canonical Correlation Analysis (CCA) [24] is a tool designed for data integration. It is recommended in a recent benchmark evaluation [25] due to the algorithm's good performance results on the batch-effect correction.

CellHeap combines UMAP [29] and PCA [30] algorithms for dimensionality reduction and cluster visualization. UMAP is a nonlinear reduction method that balances the capture of local similarity and the preservation of global data structure. A clustering method is provided by Seurat [31], which assigns cells to clusters based on their PCA scores.

Tools currently available for cell-type annotation include `Cellassign` [32], `ScType` [33], `SingleR` [34], and `scCATCH` [35].

Phase 5 - Advanced cell-level and gene-level analysis

The cell-level analysis focuses on the identification, characterization, and dynamics inference of groups of cells. The gene-level analysis focuses on the analysis of gene set expression among groups of cells.

The CellHeap platform includes two methods for cell-level analysis, trajectory analysis [36, 37, 38], and cell-cell communication [39, 40, 41, 42, 43, 44].

The CellHeap platform allows the implementation of several methods for gene-level analysis, such as differentially expressed genes (DEG) [26], pathway analysis [45], and metabolic analysis [46].

A case study of CellHeap: a COVID-19 scRNA-seq workflow

High performance computational platforms such as supercomputers are adequate for scRNA-Seq data analysis which requires a lot of storage and processing capacity.

This section describes a case study in which a CellHeap workflow was deployed in the Santos Dumont (SDumont) supercomputer environment using COVID-19 scRNA-seq datasets. This case study aimed to analyse the modulations of Fc receptors in lung lymphoid and myeloid cells imposed by COVID-19 by processing bronchoalveolar lavage fluid (BALF) samples. We found particular modulations considering mild and severe cases.

We focused on deploying the first four phases of the workflow, which are enough to justify using high-performance environments and obtain relevant biological results. We observed that the first two phases require longer execution time and greater memory capacity when compared to the other phases, as described below.

First, we describe a sequential version of CellHeap, observing the amounts of memory, storage, and execution time for each phase of this case study. Then, we propose the deployment of a parallel version of the CellHeap workflow that aims to reduce total execution time, to be detailed later.

The Santos Dumont computational environment

The Santos Dumont supercomputer (SDumont[3]) has a processing capacity of 5.1 Petaflop/s, and presents a hybrid configuration of computational nodes in its parallel processing architecture. It has 36,472 CPU cores distributed across 1,134 computing nodes, the majority of which are composed exclusively of multicore CPUs. There are two categories of computing nodes, `base` and `sequana`. The `base` nodes have 64GB RAM and are connected using an Infiniband FDR (56Gb/s) network. The `sequana` nodes have 384 GB or 768 GB RAM and are connected using an Infiniband EDR (100Gb/s) network. There are 94 GPU `sequana` nodes with NVIDIA Volta V100 accelerators and 384 GB RAM. Additionally, the SDumont has the Lustre parallel file system with a capacity of 1.7 PBytes integrated with the Infiniband network, and a secondary file system of 640 TBytes.

The SDumont Supercomputer uses the Slurm[4] as its cluster workload manager. The Slurm is an open-source, fault-tolerant, and highly scalable cluster management and job scheduling system for Linux clusters. The SDumont has twenty-six queues to submit jobs with different priorities to attend to workload characteristics.

The size of the raw sequence data of each scRNA-seq sample varies from dozens to hundreds of gigabytes. Therefore, choosing one of the execution queues depends on the raw sequence data size. Our study used the SDumont queues composed of `sequana` nodes with 384 GB RAM (sequana_cpu queue), or 768 GB RAM (sequana_cpu_bigmem queue).

Sequential Execution of CellHeap

This section lists the main tools and libraries used in our case study. It uses the SRA toolkit in the first phase and R packages in the following phases. We also present sequential execution times on the SDumont supercomputer, which influenced several choices made in the parallel version of CellHeap.

First off, at SDumont, the large and variable-sized input samples led to the need for an accurate evaluation of memory consumption and execution time in all workflow phases. If memory capacity or execution time is misestimated, the SDumont Slurm scheduler will not execute the task to completion, killing it upon attaining the maximum execution time for the assigned queue.

[3] https://sdumont.lncc.br

[4] https://slurm.schedmd.com

We began with sequential execution of the workflow in order to estimate memory consumption and execution time as accurately as possible, taking into consideration the multithreading capacity of some tools. This way, we determined which SDumont submission queues meet requirements for both factors to process the input samples fully.

Phase 1 - Sample curation and download

In this case study, CellHeap accessed the GEO [47] repository to collect samples that meet criteria defined by the EMBL protocol. From the SRA toolkit[5], CellHeap used four tools to manipulate samples: i) `vdb-dump` to obtain information; ii) `prefetch` to download the SRA samples; iii) `vdb-validate` to verify if the downloaded samples are consistent; iv) `fasterq-dump` to convert *sra* files into *fastq* files.

The workflow received as input the bronchoalveolar scRNA-seq dataset PRJNA608742 from the NIH GEO repository[6]. This dataset was generated by Liao et al. [4] and considered the samples of 12 patients as input data. Three were collected from healthy people (the control samples), three samples were collected from patients with mild COVID-19 symptoms, and six were from patients with severe COVID-19 symptoms. However, after curation and quality control, we processed only 11 samples. Each sample had two read files for paired-end runs.

We downloaded raw data samples from SRA. The `vdb-dump`, `prefetch`, and `vdb-validate` tools have sequential implementations, and do not offer the possibility of parallelization. The `prefetch` tool downloaded the 11 raw data samples (913.1 GB) in approximately 16 hours. However, after some tests, we realized that the limit imposed on the network bandwidth came from the public repository itself.

Two scripts were submitted to run this phase: one to download the input data (`prefetch` tool) sequentially, and the other (multithreaded) to generate the `fastq` input files for Phase 2 (`fasterq-dump` tool). Due to variable input file sizes, Phase 1 processing used the cpu_big_mem queue, with 768 GB RAM, to guarantee correct processing of all tasks.

The `fasterq-dump` tool is multithreaded, and consumes significant temporary disk space in addition to the final output file. During its execution, we noted that at least 13 times the sample size was needed in disk storage to guarantee the correct completion of this phase. `Fasterq-dump` took 155 hours for the 11 samples. The total execution time was approximately 172 hours for Phase 1.

Phase 1 was the most time-consuming step of this case study. After evaluating execution times and tools, we noted that `fasterq-dump` execution might be an opportunity for parallelization.

Phase 2 - Reads mapping and quantification

In this phase, Cellheap used `cellranger count` (version 6.1.0) to process scRNA-Seq data and generates the gene-barcode matrix, demanding a large amount of storage and processing capacity. This task produced several output files, of which the `barcodes.tsv`, `matrix.mtx`, and `features.tsv`, the input files for Phase 3, are the most representative. The output datasets generated by `cellranger count` also provide quality assessment reports that display essential quality metrics (i.e., Q30).

We assembled a hybrid reference genome to align reads for this case study. The assembly of the hybrid genome is executed only once by invoking the Cellranger `mkref` function, which includes both the human genome (GRCh38 v3.0.0) and the SARS-CoV-2 genome (NC_045512v2) of the severe acute respiratory syndrome coronavirus 2 isolate genome Wuhan-Hu-1. The reference genome was built offline and was an input parameter for `cellranger count`.

Notably, Phase 2 processing was both data and computing-intensive. For example, the `cellranger count` tool required more than 104 hours to process all samples of this case study in a single computational node with 768 GB RAM. In the worst case, one control sample took more than 20 hours to be processed. Processing time could be improved by distributing computing loads among multiple nodes.

Phase 3 - Quality control

In Phase 3, CellHeap used R (version 4.2.1) [48] and the packages available through the CRAN[7] family of Internet sites providing a wide variety of statistical and graphical routines.

CellHeap performs data quality control by filtering cells. First, we used the `SoupX` [23] package version 1.6.2 to estimate ambient RNA contamination. Then, we used the `Seurat` [22] package to filter cells by removing cell barcodes according to the three criteria commonly used for scRNA-Seq quality control processing [26]: UMI counts; genes expressed per cell; and percentage of mitochondrial DNA. The filtering parameters must be refined based on the tissue and the scientific question under investigation. After filtering, we remove cells contaminated with the SARS-CoV-2 virus and use only non-contaminated cells in the downstream analysis.

Phase 3 executed quickly in comparison to previous phases, between one and seven minutes per sample, depending on its size. For all 11 samples of the PRJNA608742 input dataset, the total sequential execution time was 26 minutes. Phase 3 required less memory than previous phases, and used the `sequana` nodes (384GB RAM) for submission queues at the SDumont. The output files were *tsv* files that serve as input to Phase 4. Parallelizing Phase 3 would offer only marginal gains to the CellHeap workflow so we decided to keep Phase 3 as a sequential workflow step.

Phase 4 - Sample integration, cell clustering, and cell annotation

In this phase of the case study, we used the following tools: `Seurat` package for data integration, dimensionality reduction, and cell clustering. In contrast, `Cellassign` (version 0.99.21) and `ScType` to identify the cell type based on a predefined set of gene markers.

`Seurat` carries out the clustering through a graph-based approach on a PCA reduction with several dozens of PCs (typically ≥ 30). Based on the Euclidean distance in the PCA space, we first constructed a KNN graph then used the Louvain algorithm to cluster the cells. Finally, we used the UMAP algorithm for cluster visualization, with the PCA reduction as input.

In Phase 4, we grouped samples depending on level of the severity. Therefore, there were three groups of integrated samples: control, mild and severe, with specific scripts for each group. Phase 4 scripts require a submission queue to nodes with 768 GB RAM. Groups can be processed concurrently.

Phase 4 had a longer execution time than Phase 3. Depending on the total number of cells per integrated condition,

[5] https://github.com/ncbi/sra-tools

[6] https://www.ncbi.nlm.nih.gov/Traces/study/?acc=PRJNA608742

[7] https://cran.r-project.org/

execution time varied from over 11 hours to less than 1 hour. The total sequential time was over 17 hours, and there is room for improvement by processing different conditions simultaneously.

Parallel execution of CellHeap

In our design, we devised a parallel architecture that considers many computing nodes. In this way, we can explore multinode processing to accelerate the time-consuming CellHeap phases/tools.

In Phases 1 through 3 of the sequential approach, each sample is computed by a single node, using the linear workflow shown in Figure 2. If, for instance, there are two samples, sample 1 will be fully processed before the processing of sample 2 begins.

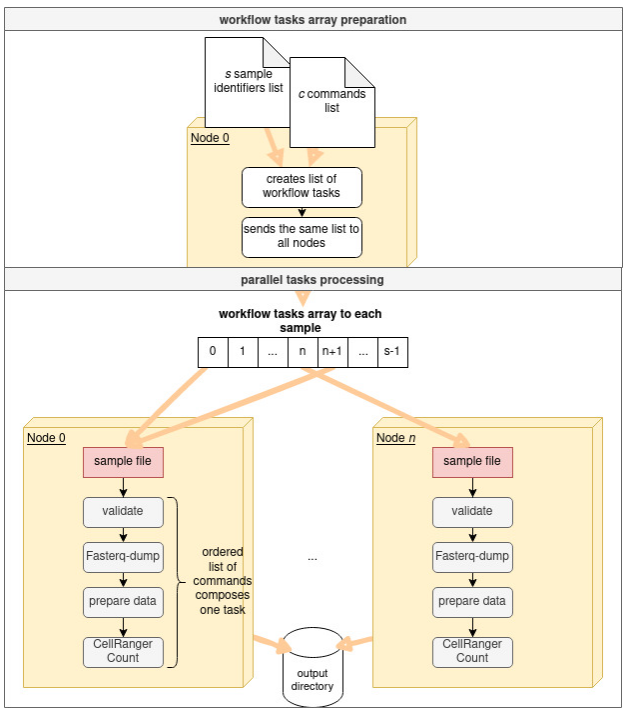

Fig. 2. Automated parallel tool to parallelize Phases 1 and 2 of CellHeap.

However, since the processing of each sample in Phases 1 through 3 is independent of the others (Bag-of-Tasks - BoT), the samples may be processed in parallel. We can therefore exploit more parallelism in the execution of the workflow.

Phases 1 and 2 consume the most memory storage and processing time in CellHeap. These phases, when parallelized, can provide substantial performance gains. Our parallel approach focuses on the tools used in these phases, e.g., `fasterq-dump` and `cellranger count`.

To distribute the computation of different samples to distinct nodes, we took into account the memory and execution time limitations. Since the execution time is proportional to the sample size, we sorted the samples by size and generated a list of sample ids, organized from largest to smallest.

Using this list of samples, we created a peer approach to process the samples simultaneously, as shown in Figure 2. In our approach, all nodes receive the same list of samples at the beginning of the computation and each node uses its node number to process the samples independently, avoiding synchronization. Our approach is lock-free, with no wait time for the list of samples, since each node always accesses a different element of the list. Each node calculates a hash using the function $s_i \mod N = n_i$, where s_i is the sample number, N is the total number of nodes, n_i is the node number, and $\mod$ returns the remainder when dividing s_i by N. This function is computed in each node to access and process its assigned samples, with no communication. In Figure 2, we show n nodes. Assuming that there are 2 nodes and that the number of samples s is equal to 8, nodes 0 and 1 will process samples $(0, 2, 4, 6)$ and $(1, 3, 5, 7)$, respectively. Each node will process a linear subset of the workflow (from `validate` to `cellranger count`).

We developed an automated parallel tool that uses MPI (Message Passing Interface) to start the execution in the nodes and send the list of samples to each node at the beginning of the computation. Then, the nodes use the local list to obtain their assigned samples and process them. At the end of the computation, the nodes write the output in a shared directory. It is important to note that, in addition to being compute-intensive, `fasterq-dump`, `gzip` and `cellranger count` also execute a great number of I/O operations in the shared file system.

Results and Discussion

In this section, we first discuss the obtained results related to execution time and memory. Then, we interpret the results from the point of view of their biological implications.

Computational Results

Table 1 shows the total time obtained when executing the sequential version of the CellHeap platform, considering the 11 samples downloaded from PRJNA608742, as previoulsy described. The size of all SRA files for 11 samples was 913.1 GB. The amount of storage required for processing is multiplied more than 10 times during Phase 2 execution.

Table 1. Execution time of the sequential version of the workflow

Phase	Execution time (hh:mm)
Phase 1	172:13
Phase 2	104:18
Phase 3	00:26
Phase 4	17:15
Total time	294:12

Phases 1, 2, and 3 are executed only once. However, Phase 4 may be executed several times to adjust parameters and answer the biological question correctly. Table 1 shows the time for a single run of Phase 4.

It is important to note that a single execution of the workflow required approximately 294 hours, or 12 days, of intensive processing, considering only the 11 samples of our case study. Furthermore, the execution times do not include wait time in SDumont's Slurm job submission queue.

To show the potential for parallelization in our COVID-19 CellHeap workflow, we focused on Phases 1 and 2, which are the most computing and data-intensive. Moreover, we did not parallelize the execution of the `prefetch` tool since its

performance seems to be defined by the NCBI public repository download policy. In this section, we focused on the six severe samples since they are large enough to be representative. In addition, we organized sample based on size to distribute the samples among the processing nodes efficiently.

Table 2 shows the name, size, and sequential running time of the six severe case samples for Phases 1 and 2. The tools used in these phases are `validate`, `fasterq-dump`, `gzip`, and `cellranger count`. This table shows that sequentially processing the severe samples takes about five days and requires substantial RAM.

Table 2. The size and sequential execution time of the six severe samples

Sample name	Size Gb	Execution Time (hh:mm)
SRR11537950	40.8	09:22
SRR11537951	46.2	10:52
SRR11537949	46.3	10:53
SRR11181959	85.7	28:39
SRR11181958	90.2	29:40
SRR11181956	95.9	30:37
Total	405.1	120:03

To simultaneously process the samples in Phase 1 (`validate`, `fastq-dump`, `gzip`) and Phase 2 (`cellranger count`) in a balanced way considering memory consumption and running time, we first allocated three nodes for execution and, since there were six samples, each node processed two samples. As explained earlier, the samples were sorted by their size and assigned to the nodes in a striped pattern, with a hash function. Thus, nodes 0, 1 and 2 computed samples $(0, 3)$, $(1, 4)$ and $(2, 5)$, respectively. Then, we assigned six nodes for execution, each of which computed one sample. As a result, the execution times to process the severe samples (Phases 1 and 2) sequentially and in parallel (3 and 6 nodes) are shown in Table 3.

Table 3. Execution time of the six severe samples (Phases 1 and 2)

Type	Execution Time (hh:mm)	Speedup
Sequential	120:03	1
Parallel (3 nodes)	46:18	2.59x
Parallel (6 nodes)	35:56	3.34x

There is a substantial reduction in total execution time when jobs are triggered simultaneously in the parallel version. For example, processing the six samples with three nodes required one day, 22 hours, and 18 minutes. On the other hand, sequential processing of the same samples took five days and 3 minutes, achieving a speedup of 2.59x, an impressive reduction of more than three days in processing time. When we used six nodes, the execution time was one day, 11 hours, and 16 minutes, reducing execution time by 11 hours as compared to execution with three nodes. These results demonstrate the great advantage of using our parallel tool in specific phase of CellHeap.

Phase 4 poses another opportunity for parallelization, with the control, severe, and mild case sample groups being executed several times depending on the biological aspect under analysis. In this case, we executed each group (severe, mild, control) as separate jobs, as shown in Table 4. In this table, in addition to the group, we show the execution time and the number of cells processed. Note that the running time is proportional to the number of cells.

Table 4. Execution time of the sequential version of the workflow by group (Phase 4)

Job	Group (number of samples)	Execution Time (hh:mm)	Number of cells
1	Severe (6)	11:19	40,733
2	Mild (2)	00:41	2,486
3	Control (3)	05:15	18,585
	Total Exec Time (sequential)	17:15	
	Total Exec Time (parallel)	11:19	

In Phase 4, we observe a reduced execution time when jobs are run in parallel. In this case, we reduced the execution time from 17:15 to 11:19 hours, achieving a speedup of 1.52x. Therefore, execution time was reduced by about 6 hours (Table 4).

Each group's time presented in Table 4 corresponds to a single execution. However, in many cases, more than one run is needed, using different parameters, and each one of those runs benefits from parallelism.

Biological results

We analyzed transcriptional modulations of Fc receptors (FcR) induced by the SARS-CoV-2 infection to test and validate our tool. FcRs bind to the Fc portion of immunoglobulins (antibodies) (Figure 3A), with some members binding to IgG ($Fc\gamma R$), IgA ($Fc\alpha R$), or IgE ($Fc\varepsilon R$), for example, [49]. Engaging the Fc domain of IgG (Figure 3A) with members of the $Fc\gamma R$ family mediates most cellular functions induced by this antibody class. This interaction is critical because many IgG-based alternative therapies to treat COVID-19 are under investigation. These include the injection of neutralizing monoclonal antibodies against multiple inflammatory mediators and the passive transfer of polyclonal plasma from convalescent patients [50].

The interaction of IgG (Figure 3A) with activation members of the $Fc\gamma R$ family [51] leads to the degranulation of inflammatory mediators, cytokines release, and much more. Therefore, we analyzed macrophages from bronchoalveolar lavage fluid (BALF) obtained from healthy individuals and COVID-19 patients with mild or severe clinical manifestations. The macrophages were identified based on the simultaneous transcription of the CD14, CD68, STAT1, TNF, IL6, IRF5, and IL1B genes. Then, we analyzed the transcription of some FcRs, intracellular molecules involved in FcR signaling pathways, and cytokines/chemokines. The violin plots of *FCAR, FCGR1A, FCGR2A, FCGR3A*, and *FCGR3B* transcription (Figure 3B) showed upregulation at the transcriptional level in macrophages from severe cases over the other groups (Figure 3B).

When we arranged our findings in a coherent cellular response (Figure 3C), using the $Fc\gamma R1A$, we observed a scenario compatible with a prominent inflammatory response in severe cases. In Figure 3C, we represented in red the molecules upregulated in severe cases compared to macrophages

from the other groups. After ligand engagement, members of the SRC family kinases are activated [52], and in our analysis, we observed the upregulation of LYN and HCK. These kinases then phosphorylate the intracellular tail of $Fc\gamma R$-associated γ chains of $Fc\gamma RIA$ (Figure 3C) [53], leading to the recruitment and activation of SYK [54], which was transcribed but not upregulated in our analysis (Figure 3C). SYK then phosphorylates many target proteins, including the PI3K, recruited to the membrane [55]. Moreover, SYK [56] and HCK [57] are essential in assembling NLRP3 inflammasome, a pivotal inflammatory structure [58]. These initial membrane-associated events trigger multiple inflammatory pathways whose components were upregulated in our analysis, like NF-kB and NLRP3 [59]. The upregulation of these pathways is compatible with the increased transcription of inflammatory mediators like TNF, IL-1β, CCL5, CXCL10, CXCL8, CCL2, IL-8 (Figure 3C), and others. We plan to analyze more FcR family members in other myeloid and extend to lymphoid cells to understand better the FcRs' impact on the SARS-CoV-2 infection.

Conclusion

This paper presents CellHeap, a flexible, extensible, and portable platform for scRNA-seq data processing deployable in supercomputers. The number of bioinformatics tools for scRNA-seq has increased substantially in recent years. Furthermore, the amount of scRNA-seq data available for analysis in public repositories is growing rapidly. Therefore, workflows, like the ones produced by CellHeap, are paramount to processing this massive amount of data, and to increasing biological knowledge.

We expect CellHeap to improve continuously, aggregating new tools as they are validated and available. We also expect to soon deploy CellHeap in other high-throughput platforms, such as computing clouds.

Key points

- This paper presents CellHeap, a flexible, portable, and robust platform for analyzing large scRNA-seq datasets.
- This paper presents the deployment of CellHeap in the Brazilian Santos Dumont supercomputer to process dozens of Terabytes of COVID-19 scRNA-seq raw data.
- This paper presents a novel solution for allocating parallel computational resources, improving total execution time.
- The results produced by the CellHeap deployment, focusing on particular modulations of Fc receptors considering mild and severe cases, verified to be coherent from a biological perspective.

Competing interests

No competing interest declared.

Author contributions statement

V. S. S., M. O. C. C., M. C. S. C., H. S. S. and M. T. S. conducted the implementation of the workflow, V. S. S., M. O. C. C., M. C. S. C., H. S. S., F. A. B. S. conceived and conducted the experiments, K. A. C. O., M. F. N., A. C. C. C. and A. H.-P. analyzed the results, M. E. M. T. W., A. C. M. A. M. and F. A. B. S. wrote the manuscript. All the authors reviewed the manuscript.

Acknowledgments

The authors acknowledge the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing SDumont supercomputer HPC resources, which have contributed to the research results reported within this paper. URL: http://sdumont.lncc.br. The authors also acknowledge the INOVA-FIOCRUZ program (grant number VPPCB-005-FIO-20-2-34-52) for funding this research. M.E.M.T.Walter and M.F.N thank CNPq for the research scholarship PQ 306947/2021-8 and PQ 306894/2019-0, respectively.

References

1. Tian Y, Carpp LN, and Miller HER et al.. Single-cell immunology of SARS-CoV-2 infection. *Nature Biotechnology*, 40:30–41, 2022.
2. Kuchina A, Brettner LM, and Paleologu L et al.. Microbial single-cell rna sequencing by split-pool barcoding. *Science*, 2020.
3. Vigneron A, O'Neill MB, and Weiss BL et al.. Single-cell rna sequencing of trypanosoma brucei from tsetse salivary glands unveils metacyclogenesis and identifies potential transmission blocking antigens. *Proceedings of the National Academy of Sciences*, 117(5):2613–2621, 2020.
4. Liao M, Liu Y, and Yuan J et al.. Single-cell landscape of bronchoalveolar immune cells in patients with covid-19. *Nature medicine*, 26(6):842–844, 2020.
5. Carangelo G, Magi A, and Semeraro R. From multitude to singularity: Na up-to-date overview of scRNA-seq data generation and Analysis. *Frontiers in Genetics*, 13:994069, 2022.
6. Zhang B, Moorlag SJ, and Dominguez-Andres J et al.. Single-cell RNA sequencing reveals induction of distinct trained-immunity programs in human monocytes. *The Journal of Clinical Investigation*, 132(7):e147719, 2022.
7. Hong R, Koga Y, and Bandyadka S et al.. Comprehensive generation, visualization, and reporting of quality control metrics for single-cell RNA sequencing data. *Nature Communications*, 13(1688):1–9, 2022.
8. Van de Sande B, Flerin C, and Davie K et al.. A scalable SCENIC workflow for single-cell gene regulatory network analysis. *Nature Protocols*, 15:2247–2276, 2020.
9. Shomroni O, Sitte M, and Schmidt J et al.. A novel single-cell RNA-sequencing approach and its applicability connecting genotype to phenotype in ageing disease. *Scientific Reports*, 12(4091):1–14, 2022.
10. Aalst WMP. Flexible Workflow Management Systems: An Approach Based on Generic Process Models. In *Proceedings of the Database and Expert Systems Applications (DEXA)*, pages 186–195, 1999.
11. Deelman E, Peterka T, and Altintas I et al.. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, 32(1):159–175, 2018.
12. Liu W, Jia J, and Dai Y et al.. Delineating COVID-19 immunological features using single-cell RNA sequencing. *The Innovation*, 3(5):100289, 2022.

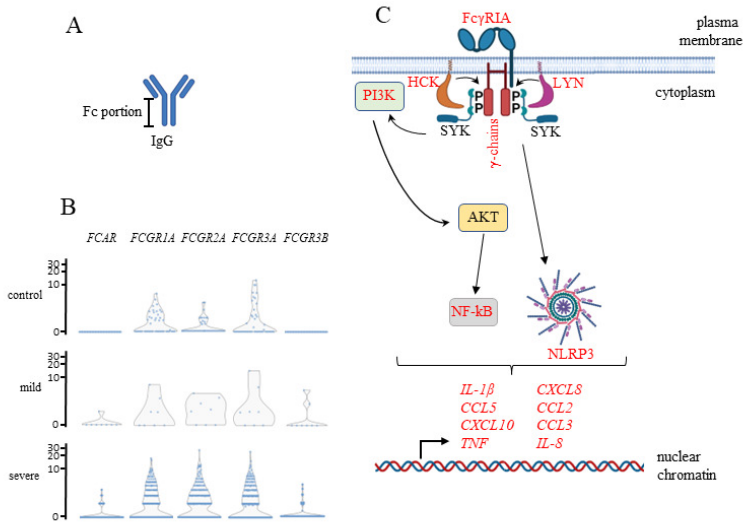

Fig. 3. *Analysis of Fc receptors in macrophages from COVID-19 patients.* The biological analysis was performed in bronchoalveolar lavage fluid (BALF) macrophages obtained from healthy (control) or COVID-19 patients with mild or severe symptoms. A) shows a schematic representation of an IgG molecule with its Fc portion. B) violin plots of the indicated FcRs for the three groups of patients. C) alignment of analyzed molecules that participate in $Fc\gamma R1A$ response according to known intracellular pathways. All molecules represented in red were upregulated in severe COVID-19 cases compared to the other groups.

13. Aznaourova M., Schmerer N, and Janga H et al.. Single-cell RNA sequencing uncovers the nuclear decoy lincRNA PIRAT as a regulator of systemic monocyte immunity during COVID-19. *PNAS*, 119(36):1–12, 2022.
14. Schulte-Schrepping J, Reusch N, and Paclik D et al.. Severe covid-19 is marked by a dysregulated myeloid cell compartment. *Cell*, 182(6):1419–1440, 2020.
15. Yao C, Bora SA, and Parimon T et al.. Cell-type-specific immune dysregulation in severely ill covid-19 patients. *Cell Reports*, 34(1), 2020.
16. Song E, Bartley CM, and Chow RD. Divergent and self-reactive immune responses in the cns of covid-19 patients with neurological symptoms. *Cell Reports Medicine*, 2(5), 2021.
17. Silvin A, Chapuis N, and Dunsmore G et al.. Elevated calprotectin and abnormal myeloid cell subsets discriminate severe from mild covid-19. *Cell*, 182(6), 2020.
18. Guo C, Li B, and Ma H et al.. Tocilizumab treatment in severe COVID-19 patients attenuates the inflammatory storm incited by monocyte centric immune interactions revealed by single-cell analysis. *Nature Communications*, 11:3924, 2020.
19. Jørgensen MJ, Holter JC, and Christensen EE et al.. Increased interleukin-6 and macrophage chemoattractant protein-1 are associated with respiratory failure in covid-19. *Scientific reports*, 10(1):1–11, 2020.
20. Stephenson E, Reynolds G, and Botting RA et al.. Single-cell multi-omics analysis of the immune response in COVID-19. *Nature Medicine*, 27:904–916, 2021.
21. Zhang X, Li T, and Liu F et al.. Comparative analysis of droplet-based ultra-high-throughput single-cell rna-seq systems. *Molecular cell*, 73(1):130–142, 2019.
22. Hao Y, Hao S, and Andersen-Nissen E et al.. Integrated analysis of multimodal single-cell data. *Cell*, 2021.
23. Young MD and Behjati S. Soupx removes ambient rna contamination from droplet-based single-cell rna sequencing data. *Gigascience*, 9(12):giaa151, 2020.
24. Stuart T, Butler A, and Hoffman P et al.. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.
25. Tran HTN, Ang KS, and Chevrier M et al.. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21(1):1–32, 2020.
26. Luecken MD and Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(e8746):1–23, 2019.
27. Heimberg G, Bhatnagar R, and El-Samad H et al.. Dimensionality in Gene Expression Data Enables the Accurate Extraction of Transcriptional Programs from Shallow Sequencing. *Cell Systems*, 2(4):239–250, 2016.
28. Baran Y, Bercovich A, and Sebe-Pedros A et al.. Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome biology*, 20(1):1–19, 2019.
29. Becht E, McInnes L, and Healy J et al.. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.
30. Ringnér M. What is principal component analysis? *Nature biotechnology*, 26(3):303–304, 2008.

31. Hao Y, Hao S, and Andersen-Nissen E et al.. Integrated analysis of multimodal single-cell data. *Cell*, 2021.
32. Zhang AW, O'Flanagan C, and Chavez EA et al.. Probabilistic cell-type assignment of single-cell rna-seq for tumor microenvironment profiling. *Nature methods*, 16(10):1007–1015, 2019.
33. Ianevski A, Giri AK, and Aittokallio T. Fully-automated and ultra-fast cell-type identification using specific marker combinations from single-cell transcriptomic data. *Nature communications*, 13(1):1–10, 2022.
34. Aran D, Looney AP, and Liu L et al.. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology*, 20(2):163–172, 2019.
35. Shao X, Liao J, and Lu X et al.. sccatch: automatic annotation on cell types of clusters from single-cell rna sequencing data. *Iscience*, 23(3):100882, 2020.
36. Herring CA, Banerjee A, and McKinley ET et al.. Unsupervised Trajectory Analysis of Single-Cell RNA-Seq and Imaging Data Reveals Alternative Tuft Cell Origins in the Gut. *Cell Systems*, 6(1):37–51, 2018.
37. Street K, Risso D, and Fletcher R et al.. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(477):1–16, 2018.
38. Wolf FA, Hamey FK, and Plass M et al.. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biology*, 20(59):1–9, 2019.
39. Efremova M, Vento-Tormo M, and Teichmann SA et al.. Cellphonedb: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature protocols*, 15(4):1484–1506, 2020.
40. Shao X, Liao J, and Li C et al.. Celltalkdb: a manually curated database of ligand–receptor interactions in humans and mice. *Briefings in bioinformatics*, 22(4):bbaa269, 2021.
41. Noël F, Massenet-Regad L, and Carmi-Levy I et al.. Dissection of intercellular communication using the transcriptome-based framework icellnet. *Nature communications*, 12(1):1–16, 2021.
42. Palla G, Spitzer H, and Klein M et al.. Squidpy: a scalable framework for spatial omics analysis. *Nature methods*, 19(2):171–178, 2022.
43. Jin S, Guerrero-Juarez CF, and Zhang L et al.. Inference and analysis of cell-cell communication using cellchat. *Nature communications*, 12(1):1–20, 2021.
44. Cillo AR, Kürten CHL, and Tabib T et al.. Immune landscape of viral-and carcinogen-driven head and neck cancer. *Immunity*, 52(1):183–199, 2020.
45. Fabregat A, Jupe S, and Matthews L et al.. The Reactome Pathway Knowledgebase. *Nucleic Acids Research*, 4(46(D1)):D649–D655, 2018.
46. Damiani C, Maspero D, and Di Filippo M et al.. Integration of single-cell rna-seq data into population models to characterize cancer metabolism. *PLoS computational biology*, 15(2):e1006733, 2019.
47. Clough E and Barrett T. The gene expression omnibus database. *Methods in molecular biology*, 1418:93–110, 2016.
48. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.
49. Daëron M. Fc receptors as adaptive immunoreceptors. *Fc Receptors*, pages 131–164, 2014.
50. van de Veerdonk FL, Giamarellos-Bourboulis E, and Pickkers P et al.. A guide to immunotherapy for covid-19. *Nature Medicine*, 28(1):39–50, 2022.
51. Bournazos S, Gupta A, and Ravetch JV. The role of igg fc receptors in antibody-dependent enhancement. *Nature Reviews Immunology*, 20(10):633–643, 2020.
52. Suzuki T, Kono H, and Hirose N et al.. Differential involvement of src family kinases in fcγ receptor-mediated phagocytosis. *The Journal of Immunology*, 165(1):473–482, 2000.
53. Daëron M. Fc receptor biology. *Annual review of immunology*, 15:203, 1997.
54. Mócsai Aand Ruland J and Tybulewicz VLJ. The syk tyrosine kinase: a crucial player in diverse biological functions. *Nature Reviews Immunology*, 10(6):387–402, 2010.
55. Klippel A, Reinhard C, and Kavanaugh WM et al.. Membrane localization of phosphatidylinositol 3-kinase is sufficient to activate multiple signal-transducing kinase pathways. *Molecular and cellular biology*, 16(8):4117–4127, 1996.
56. Neumann K and Ruland J. Kinases conquer the inflammasomes. *Nature immunology*, 14(12):1207–1208, 2013.
57. Kong X, Liao Y, and Zhou L et al.. Hematopoietic cell kinase (hck) is essential for nlrp3 inflammasome activation and lipopolysaccharide-induced inflammatory response in vivo. *Frontiers in pharmacology*, 11:581011, 2020.
58. Petrilli V, Papin S, and Tschopp J. The inflammasome. *Current Biology*, 15(15):R581, 2005.
59. Liu T, Zhang L, and Joo D et al.. Nf-κb signaling in inflammation. *Signal transduction and targeted therapy*, 2(1):1–9, 2017.

Maria Clicia S. Castro obtained her DSc from the Systems and Computing Engineering Program at COPPE/UFRJ, Brazil, in 1998. She is a Full Professor at the Department of Informatics and Computer Science at the Rio de Janeiro State University, Brazil. She is also a Prociência Research Fellow. Prof. Castro has been working for the last 32 years in high-performance computing. In 2021, she did a sabbatical at the Faculté d'Ingénierie, Sorbonne Université. Her research interests are high-performance computing, bioinformatics, and cloud computing.

Vanessa S. Silva has a postdoctoral degree in Bioinformatics and a Ph.D. in Computational Biology and Systems from Oswaldo Cruz Foundation (Fiocruz/Brazil) with experience in bioinformatics, transcriptomics, and molecular modeling in drug design, acting on the following subjects: scRNA-seq analysis, Big Data processing, and molecular dynamics.

Maiana O. C. Costa obtained her D.Sc. in Computational Modeling from the National Laboratory of Scientific Computation (LNCC), Brazil, in 2018. Since 2019, Maiana Costa is a postdoctoral researcher at the National Laboratory of Scientific Computation (LNCC) working in Bioinformatics and Systems Biology research. Her research interests are genomics, transcriptomics, and metabolic and regulatory networks applied to clinical pathogens.

Helena S. Silva is working toward her Master's degree in Computer Science at the University of Brasilia, Brazil, under the supervision of Prof. Alba C. M. A. Melo. Her research interests are high-performance computing, scientific workflows and distributed computing.

Maria Emilia M. T. Walter obtained her Doctorate from the University of Campinas, Brazil, in 1999. Since 1988, she works at the Department of Computer Science at the University of Brasilia, Brazil, where she is Adjunct Professor. She is Brazilian CNPq Researcher level 2 and, since 2017, she is the Dean of Research and Innovation of the University of Brasilia. Her research interests are high-performance sequencing projects, comparative genomics, identification and classification of non-coding RNAs and genome rearrangements.

Alba C. M. A. Melo obtained her PhD from INPG (Institut National Polytechnique de Grenoble), France, in 1996. Since 1997, she works at the Department of Computer Science at the University of Brasilia, Brazil, where she is now Full Professor. Prof. Melo is Senior Member of the IEEE (Institute of Electrical and Electronic Engineers), Latin America Coordinator of the IEEE TCPP (Technical Community of Parallel Processing), Member of the BRICS (Brazil, Russia, India, China and South Africa) Working Group on High Performance Computing and Information Communication Technologies, Member of the Council of the Brazilian Computer Society and Brazilian CNPq Researcher level 1C. Her research interests are high performance computing, bioinformatics and cloud computing.

Kary A. C. Ocaña is a Full Researcher at Bioinformatics Laboratory of LNCC. From 2010 to 2015, she worked as a postdoctoral researcher at the Department of Computer Science, COPPE Institute, UFRJ, Brazil. She received both her D.Sc. (2010) and M.Sc. (2006) in Cellular and Molecular Biology from FIOCRUZ (RJ-Brazil). She participates in pharmaceutical biochemistry and molecular and cell biology research projects with funding from government agencies. Her research interests include phylogenomics, the development of bioinformatics software and workflows, HPC, data analytics, and machine learning.

Marcelo T. dos Santos obtained his DSc degree in Theoretical Physics from the University of São Paulo, Brazil, in 1997. He is a Technologist at the National Scientific Computing Laboratory (LNCC) since 2002, his research interest is mainly Computational Systems Biology.

Marisa F. Nicolas. She obtained her Ph.D. in Genetics from the Federal University of Paraná, Brazil, in 2004. Marisa Nicolás is an associate researcher at the National Scientific Computing Laboratory, LNCC working in genomics and bioinformatics research. Her research interests are genomic, transcriptomic, metabolic, and regulatory networks in studies with clinical pathogens.

Anna Cristina C. Carvalho is a physician, graduated in Medicine at the Federal University of Rio de Janeiro (UFRJ), specialist in Infectious Diseases (UFRJ), with PhD degree in Brazil (UFRJ) and Italy (Universita degli Studi di Brescia). Since 2013 works as Public Health Researcher at the Oswaldo Cruz Institute (IOC), Fiocruz, in Rio de Janeiro, Brazil, where she develops clinical and epidemiological studies on global health and neglected infectious diseases, mainly on tuberculosis. She teaches in postgraduate courses at UFRJ and Fiocruz, is the institutional cooperation advisor at IOC, coordinator member of the Brazilian Tuberculosis Research Network (REDE-TB), deputy editor of the International Journal of Tuberculosis and Lung Disease (IJTLD) and a senior scientist from CNPq (Brazil).

Andrea Henriques-Pons has a degree in Biological Sciences (Bachelor's Degree in Genetics) (1990), a Master's degree in Biophysics (1995), and a Ph.D. in Biophysics, all from the Federal University of Rio de Janeiro (2000). She has a postdoctoral degree in stem cell biology from Children's Hospital in Washington DC. She is currently a Senior Researcher at Fundação Oswaldo Cruz, Instituto Oswaldo Cruz, with experience in cell biology, with an emphasis on Immunology, working mainly on the following topics: Trypanosoma cruzi infection, flow cytometry, inflammation, myocarditis, and Duchenne muscular dystrophy.

Fabricio A. B. Silva obtained his PhD from Université Pierre et Marie Curie, France, in 2000. He was a visiting scholar at the University of California Irvine in 2001 and visiting professor at the University of Lisbon from 2008 to 2010. Since 2013, Fabricio Silva is a Senior Researcher in Public Health at Oswaldo Cruz Foundation, Rio de Janeiro, Brazil. His research interests are systems biology, computational biology, and large-scale analysis of biological data.