





Advancing HIV Vaccine Research With Low-Cost High-Performance Computing Infrastructure: An Alternative Approach for Resource-Limited Settings

Bioinformatics and Biology Insights
Volume 13: 1–8
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1177932219882347



Batsirai M Mabvakure^{1,2,3}, Raymond Rott⁴, Leslie Dobrowsky⁴, Peter Van Heusden⁵, Lynn Morris^{1,2,6}, Cathrine Scheepers^{1,2} and Penny L Moore^{1,2,6}

¹Center for HIV and STIs, National Institute for Communicable Diseases, National Health Laboratory Service (NHLS), Johannesburg, South Africa. ²Antibody Immunity Research Unit, Faculty of Health Sciences, University of the Witwatersrand, Johannesburg, South Africa. ³Division of Transfusion Medicine, Department of Pathology, Johns Hopkins University School of Medicine, Baltimore, MD, USA. ⁴Bridge-the-Gap, Johannesburg, South Africa. ⁵South African National Bioinformatics Institute, University of the Western Cape, Cape Town, South Africa. ⁶Centre for the AIDS Programme of Research in South Africa (CAPRISA), University of KwaZulu-Natal, Durban, South Africa.

ABSTRACT: Next-generation sequencing (NGS) technologies have revolutionized biological research by generating genomic data that were once unaffordable by traditional first-generation sequencing technologies. These sequencing methodologies provide an opportunity for in-depth analyses of host and pathogen genomes as they are able to sequence millions of templates at a time. However, these large datasets can only be efficiently explored using bioinformatics analyses requiring huge data storage and computational resources adapted for high-performance processing. High-performance computing allows for efficient handling of large data and tasks that may require multi-threading and prolonged computational times, which is not feasible with ordinary computers. However, high-performance computing resources are costly and therefore not always readily available in low-income settings. We describe the establishment of an affordable high-performance computing bioinformatics cluster consisting of 3 nodes, constructed using ordinary desktop computers and open-source software including Linux Fedora, SLURM Workload Manager, and the Conda package manager. For the analysis of large antibody sequence datasets and for complex viral phylogenetic analyses, the cluster out-performed desktop computers. This has demonstrated that it is possible to construct high-performance computing capacity capable of analyzing large NGS data from relatively low-cost hardware and entirely free (open-source) software, even in resource-limited settings. Such a cluster design has broad utility beyond bioinformatics to other studies that require high-performance computing.

KEYWORDS: High-performance computing, bioinformatics, data analysis, large data, low-cost systems, next-generation sequencing, cluster

RECEIVED: September 4, 2019. **ACCEPTED:** September 21, 2019.

TYPE: Original Research

FUNDING: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors acknowledge research funding from the South African Medical Research Council (MRC) SHIP program and the International AIDS Vaccine Initiative (IAVI). IAVI's work is made possible by generous support from many donors including: the Bill & Melinda Gates Foundation; the Ministry of Foreign Affairs of Denmark; Irish Aid; the Ministry of Finance of Japan in partnership with The World Bank; the Ministry of Foreign Affairs of the Netherlands; the Norwegian Agency for Development Cooperation (NORAD); the United Kingdom Department for International Development (DFID), and the United States Agency for International Development

(USAID). The full list of IAVI donors is available at www.iavi.org. The contents are the responsibility of the authors and do not necessarily reflect the views of USAID or the United States Government. PLM and PvH are supported by the South African Research Chairs Initiative of the Department of Science and Technology and the NRF (Grant Nos 98341 and 64571, respectively). BMM is supported by a NRF SARChI Chair-linked PhD bursary and the Poliomyelitis Research Foundation.

DECLARATION OF CONFLICTING INTERESTS: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

CORRESPONDING AUTHOR: Penny L Moore, Center for HIV and STIs, National Institute for Communicable Diseases, National Health Laboratory Service (NHLS), Private Bag X4, Sandringham, Johannesburg 2131, South Africa. Email: pennym@nicd.ac.za

Introduction

Next-generation sequencing (NGS) technologies have revolutionized the sequencing landscape through their ability to generate millions of sequences at a time and provided new insights into pathogen and host evolution.^{1–3} Furthermore, the development of cheaper NGS systems means that this is no longer restricted to well-resourced centers of excellence and is now becoming a standard tool in many molecular biology laboratories including several African countries.⁴ For example, the Oxford Nanopore MinION costs just less than US\$1000, and each flow cell can generate 10 to 20 Gb of DNA sequence data.⁵ Although NGS technologies are getting increasingly portable and cheaper, data management is lagging behind.⁶ The amount of data generated using NGS comes with significant memory and storage requirements, making it challenging to analyze using desktop computers.

Some bioinformatics programs required for such analyses are computationally intensive, requiring high processing

power and long computational times either because of the large datasets being analyzed or the complex calculations and simulations performed.^{7,8} Clusters and servers have been tailored to perform such analyses; however, these are very expensive to establish. This means that researchers from low-income settings (where pathogen NGS data would be particularly useful to confront high burdens of disease) might not be able to afford to buy or rent the necessary computational resources. In settings where existing computational resources are available but scarce, Internet is often a limiting factor. Transfer of large datasets between high-performance computing centers and the research centers is often very expensive, or the Internet connections are too slow to be effective. In addition, some of the research data (eg, patient data) may contain personal and sensitive information that require high levels of protection, making it risky to transfer into public domains. Finally, many institutions have firewalls that prevent users from accessing outside servers and clusters to analyze their



Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (<http://www.creativecommons.org/licenses/by-nc/4.0/>) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

data, forcing researchers to perform analyses using ordinary desktop computers.

Our laboratory is focused on studies of viral-host evolution in the context of HIV infection and vaccination. Understanding the interplay between the host immune system and the evolving virus is a major aspect of HIV vaccine design. We use bioinformatics analyses of Sanger and NGS sequence data to analyze viral and antibody gene evolution. The human antibody repertoire is enormous, at greater than 10^{11} molecules per individual.⁹⁻¹² NGS technologies are, therefore, ideally suited for antibody repertoire analyses as they are able to sequence millions of antibody reads at a time. Similarly, the HIV envelope protein, which is the target of antibody responses during HIV infection, varies by as much as 30% between infected individuals.^{13,14} The amount of data generated through these types of studies has resulted in our laboratory encountering the computational challenges described above.

To enhance our analyses, we have developed a low-cost bioinformatics cluster that is easy to build, capable of analyzing large NGS datasets and performing phylodynamic analyses, and less reliant on Internet data usage. This system, which is amenable to analysis of diverse large datasets, will enhance the ability of under-resourced researchers to independently interrogate large datasets to address locally relevant scientific problems.

Methods

Cluster architecture

Development of a multi-node cluster

Computer hardware. A 3-node (12-core) cluster, consisting of a master node and 2 subsidiary nodes, was developed using ordinary personal computer workstations (Figure 1A). The master node (Bio-Linux) is an Intel(R) Xeon(R) CPU E3-1220 v3 at 3.10 GHz, with 4 CPU cores with 32GB RAM and 1TB SSD. Nodes 01 and 02 are Intel(R) Core(TM) i7-3770 CPU at 3.40 GHz machines, with 4 cores per socket and 32GB RAM (increased from an initial 4GB RAM per node). We also installed 11T and 50T network-attached storage (NAS) for raw and processed data, respectively (Figure 1A). The nodes perform the computational tasks, whereas the storage, as the name suggests, are the devices that store all the data that are generated. Files are also backed up to a separate file system that is further backed up to Linear Tape-Open (LTO) tapes that have a capacity of 3T each.

Operating system and packages. Fedora release 23 was installed on all the nodes. The standard installation of Linux was used, which does not have a graphical user interface. Fedora uses RPM (Redhat Package Manager), and the packages were installed from their repositories using either “yum install” or “dnf install.” The packages and all file components were extracted during installation and stored in the correct locations on the system (the default location being `/var/lib/rpm`). Cluster users were created using the Fedora dashboard.

Networking. Users login to the master node, which is connected to the external Internet (Figure 1A). All the other nodes are connected through a local area network (LAN) to the master node. External access to the cluster is given by ssh on a non-standard port to reduce the risk of port scanning by automated bots. A firewall was also put in place on the LAN network as part of cybersecurity. Users login by ssh on their terminal using a username and password supplied by the system administrator.

Power. The electricity from the main electrical supply passes through a generator and uninterruptible power supply (UPS) before passing through a secondary local UPS, to avoid disruption of analyses due to power failure.

Cluster configuration

SLURM. Simple Linux Utility for Resource Management (SLURM) version 15.08 was installed to manage the cluster resources. SLURM allocates exclusive and/or non-exclusive access to resources (computer nodes) to users for a defined duration of time by providing a framework for starting, executing, and monitoring jobs (normally a parallel job) on the set of allocated nodes and arbitrates contention for resources by managing a queue of pending jobs. This means users can share the cluster in a controlled manner. SLURM was first installed on the master node, and the process was repeated on the additional nodes. The number of nodes can be increased in future depending on the laboratory computational requirements. The instructions for the step-by-step approach for installing were obtained on the following website: https://slurm.schedmd.com/quickstart_admin.html.

Bioinformatics cluster file system

Data and scientific software packages are shared between nodes in the cluster using Network File System (NFS). The `/opt/conda2` directory (for scientific software) and `/home` directory (for data being analyzed) are exported from the master node and mounted on the same paths on the worker nodes. This allows scripts submitted to the SLURM scheduler to run on a consistent environment on all nodes on the cluster. User information was synchronized between nodes using Ansible 2.2.0 (<https://www.ansible.com/>).

The cluster has two storage facilities, 11T for raw data and 50T for processed data (Figure 1A). The storage for raw data is accessible to the institutional sequencing core (where the Illumina MiSeq instruments are housed) to upload data. Cluster users access the cluster from the master node and are able to access the raw data already uploaded in the storage. Users first have to copy the data from the raw data storage into the home directory and then analyze it using various bioinformatics tools installed in the `/opt` directory. Cluster users have access to the green and blue areas (Figure 1B). The red area shows an example of system files only accessible to the cluster administrator.

The user home directories are on the master node, and by default, all the jobs running write their output on the master

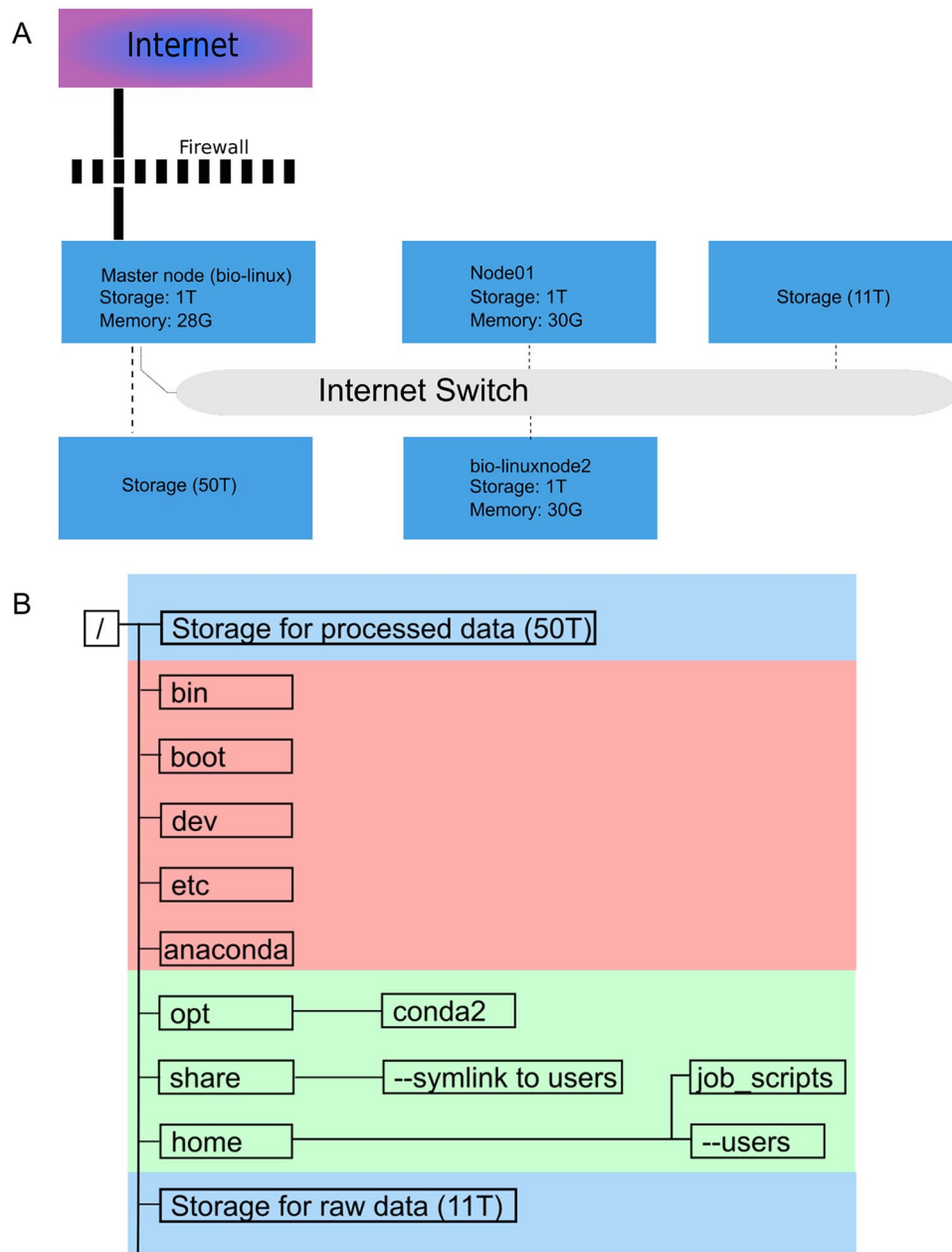


Figure 1. Bioinformatics cluster architecture. (A) Schematic description of the cluster architecture, storage, and memory. (B) Bioinformatics cluster file system. The storage area is shown in blue, the user area for data analysis in green, and the restricted system files in red.

node disk space which only has 1T capacity. Once the disk space is full, the master node would not be able to allocate other nodes to run jobs. To resolve this, we decongested the master node by mounting a 11T volume to /share to increase the storage space in the share folder. Symbolic links (symlinks) were then created for all the user home folders to the share directory to prevent the master node from becoming congested (Figure 1B).

Package manager and installation of bioinformatics packages

We installed several bioinformatics packages relevant to our studies of viral and antibody sequences described below.

However, these could be replaced with tools relevant to local laboratory needs. We installed conda (<https://conda.io/docs/intro.html>) as our package manager. All the installed packages were located in /opt/conda2/ (Figures 1 and 2). All programs were installed on the master node and are executable on all the computer nodes. Conda is a package manager application that quickly installs, runs, and updates packages and their dependencies. The conda command is the primary interface for managing installations of various packages. It can query and search the package index and current installation, create new environments, and install and update packages into existing conda environments. Creation of different environments is done for programs that might have conflicting requirements in terms of the dependencies used

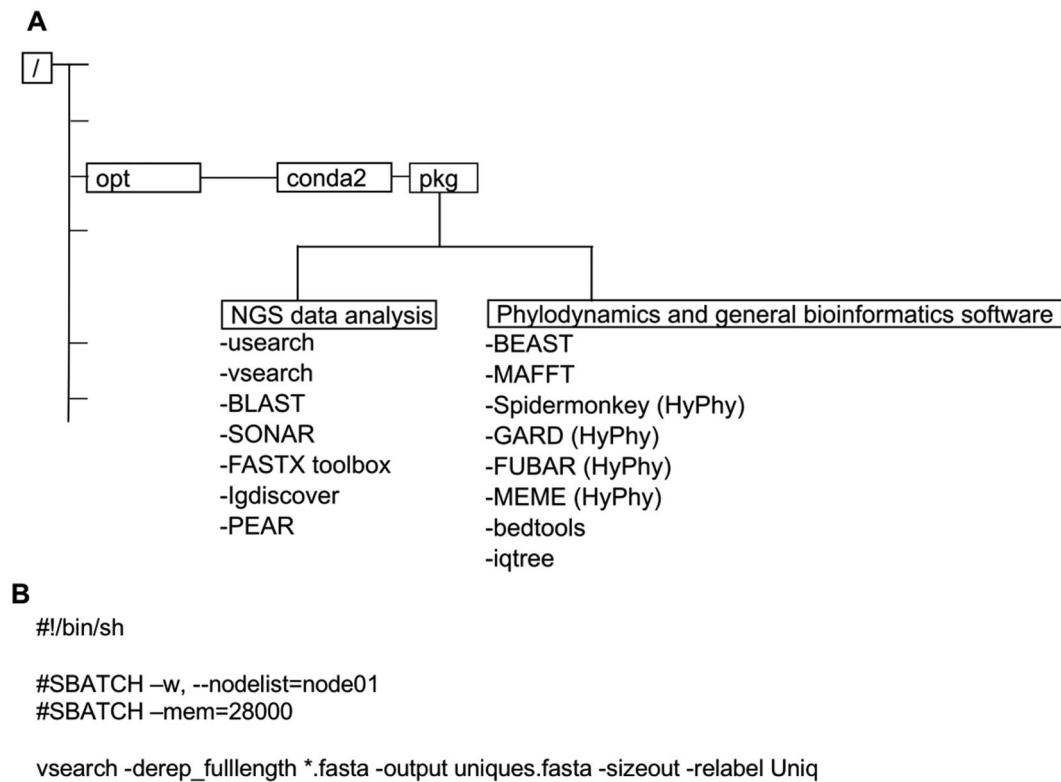


Figure 2. Access to bioinformatics programs on the cluster. (A) Bioinformatics programs on the cluster. (B) Example of a bash script to run programs on the cluster.

or versions of certain binaries (pre-built executables). The bioconda channel was enabled within conda to provide access to its collection of over 4000 open-source bioinformatics packages.¹⁵

We used open-source tools and established pipelines developed by collaborators to perform bioinformatics analyses on viral and antibody sequence data. Bash scripts were used to execute the various bioinformatics programs allowing us to specify a given node and memory requirements to run the program, an example of which is shown in Figure 2B. Job scripts are stored in a folder accessible by all users, standardizing all parameters for performing analyses and saving time when performing similar analyses between donors.

HIV-1 and antibody datasets used for the analyses

Datasets used for the analyses were collected from participants in the Center for the AIDS Program of Research in South Africa (CAPRISA).¹⁶ These individuals were followed-up from time of infection through chronic HIV infection. Ethics clearance for the use of samples was obtained from the Human Research Ethics Committee (Medical) from the University of Witwatersrand (MM040202), the University of Cape Town (025/2004), and CAPRISA at the University of KwaZulu-Natal (E013/04). HIV-1 sequences were obtained using the Sanger method,¹⁷ and antibody data from NGS with MiSeq Illumina as described previously.¹⁸

Results

Cluster performance with computationally intensive programs

Bioinformatics programs perform many analyses ranging from identifying signatures in sequences to performing complex calculations, simulations, and predictions. Some of these programs are very computationally intensive, requiring large memory and longer computational hours. These types of analyses include virus phylogenetics that provide insights into, for example, intra-host evolution of HIV-1 by estimating the rates of evolution, selection, diversity, divergence, elucidating spatio-temporal distributions of viruses, and identifying number of viral infections.^{19,20}

To achieve this, we installed a number of phylogenetics programs on the cluster, including Bayesian Evolutionary Analysis by Sampling Trees (BEAST),²¹ which uses Bayesian statistical methods that require long computational times and memory requirements. BEAST ran faster on the bioinformatics cluster compared with an ordinary machine (164 and 284 hours, respectively) to complete the same task (Table 1). The “ordinary machine” used for the comparison was a MacBook Pro with a 2.6-GHz Intel Core i7 processor and 16GB RAM. The MacBook Pro makes a good comparison with the computational cluster since it is a high-end machine with above-average processing power and memory compared with other ordinary machines. Furthermore, the bioinformatics cluster

Table 1. Comparison of the cluster performance with that of a high specification ordinary machine.

MACHINE	AVERAGE SPEED AND RANGE (HOURS/MILE STEPS)	TOTAL TIME TO COMPLETE 400 MILLION STEPS (HOURS)	NO. OF ANALYSES AT A TIME
Cluster	0.41 (0.39-0.43)	164	18
Ordinary machine: 2.8GHz Core i7, 16GB memory	0.78 (0.25-1.01)	284	1

The cluster runs jobs faster compared with the ordinary machine and also runs multiple jobs in parallel, therefore reducing the total time to complete multiple analyses.

analyzed 18 datasets at a time compared with one on the ordinary machine. This further highlights the benefits of the cluster in rapidly performing analyses with computationally intensive programs. For BEAST, data visualization was done using FigTree and SpreaD3 installed on local machines.^{22,23}

Cluster performance with large datasets

The cluster has also been applied to antibody repertoire analyses that rely on large datasets to accurately capture their evolution. We have thus far analyzed more than 3 TB of NGS antibody repertoire data (Figure 3A), which far exceeds the capability of ordinary desktop computers. These data were obtained from 4 HIV-infected participants at multiple time-points ranging between 7 and 281 weeks post infection (Figure 3B). Each analysis (2-10 million light and heavy chain antibody reads) took 8 to 168 computational hours, and the memory usage varied between 4 and 20 GB. The time and memory requirements of each of the analysis were dependent on the size of the input data, that is, the number of reads in the dataset. The cluster enables users to run several jobs simultaneously, thus allowing multiple time-points to be analyzed concurrently.

Easy integration of bioinformatics programs on the cluster

Bioinformatics analyses of complex data often involve using different tools to perform specialized tasks as part of a pipeline or a workflow. An example is the antibody repertoire NGS data analysis that we use, which requires many steps, summarized in Figure 4, and for which all of the steps are performed on the cluster. The analysis steps involve the use of the SONAR pipeline,²⁴ which links several bioinformatics tools, including Clustal Omega,^{25,26} MUSCLE,²⁵ Basic Local Alignment Search Tool (BLAST+),²⁷ BEAST,²⁸ and DNAML,²⁹ to process the data and identify sequences related to an antibody sequence of interest (clonally related sequences) (Figure 4).²⁴

The pipeline was installed as follows: dependencies were first installed using conda or as per the developer's instructions if not packaged within conda. SONAR was downloaded from <https://github.com/scharch/SONAR> and placed in `/opt/conda2/pkgs/`. After uncompressing the files, we installed the pipeline by following the command prompts after executing "setup.sh." The command prompts allowed us to specify the paths to the installed dependencies. Plots for data

visualization were made using R which was also installed on the cluster. We then defined the python path using the following command: "export PYTHONPATH=\$PYTHONPATH:/opt/conda2/pkgs/sonar/." Finally, for all the cluster users to be able to use the bioinformatics programs, we edited the `.bashrc` scripts to include the path to all the programs. The cluster, therefore, allows easy integration and automation of these bioinformatics programs, enhancing the laboratory's throughput.

Discussion

Lack of infrastructure and limited resources are bottlenecks in research conducted in low-to-medium income settings. These regions are often also those that bear the major burden of infectious diseases such as HIV. Conducting disease surveillance and vaccine research to curb the spread of locally relevant pathogens is a public health priority. Our research laboratory is involved in HIV vaccine research which involves generating and analyzing huge amounts of sequencing data. We have successfully set up a bioinformatics cluster using cheap and low-specification CPUs and demonstrated its application in analyzing these large NGS datasets. This approach has broad utility for other pathogens and beyond bioinformatics to other studies that require high-performance computing.

The cost of constructing this cluster was reduced using relatively old computers that were no longer used in the laboratory and which were upgraded at a small cost to accommodate more data. This approach is very feasible in resource-limited laboratories that have access to old computers, or funding to purchase relatively inexpensive ones. We reduced costs through the use of open-source software to configure the cluster. There is a significant body of useful open-source software that also comes with good technical support. Much of this support comes from the community of users that interact on Google groups such as Gitter, GitHub, Biostars, Stack Overflow, and other online platforms. We used Linux Fedora operating system, as it is open source, stable, and is sponsored and funded by Red Hat. Conda package manager is also open-source software that makes it easy and manageable to install bioinformatics packages. The SLURM scheduler is also open-source software with excellent community support.

There are a number of software packages that may be used for cluster configuration management such as Ansible, Chef, and Puppet. Puppet has advantages when it comes to updating

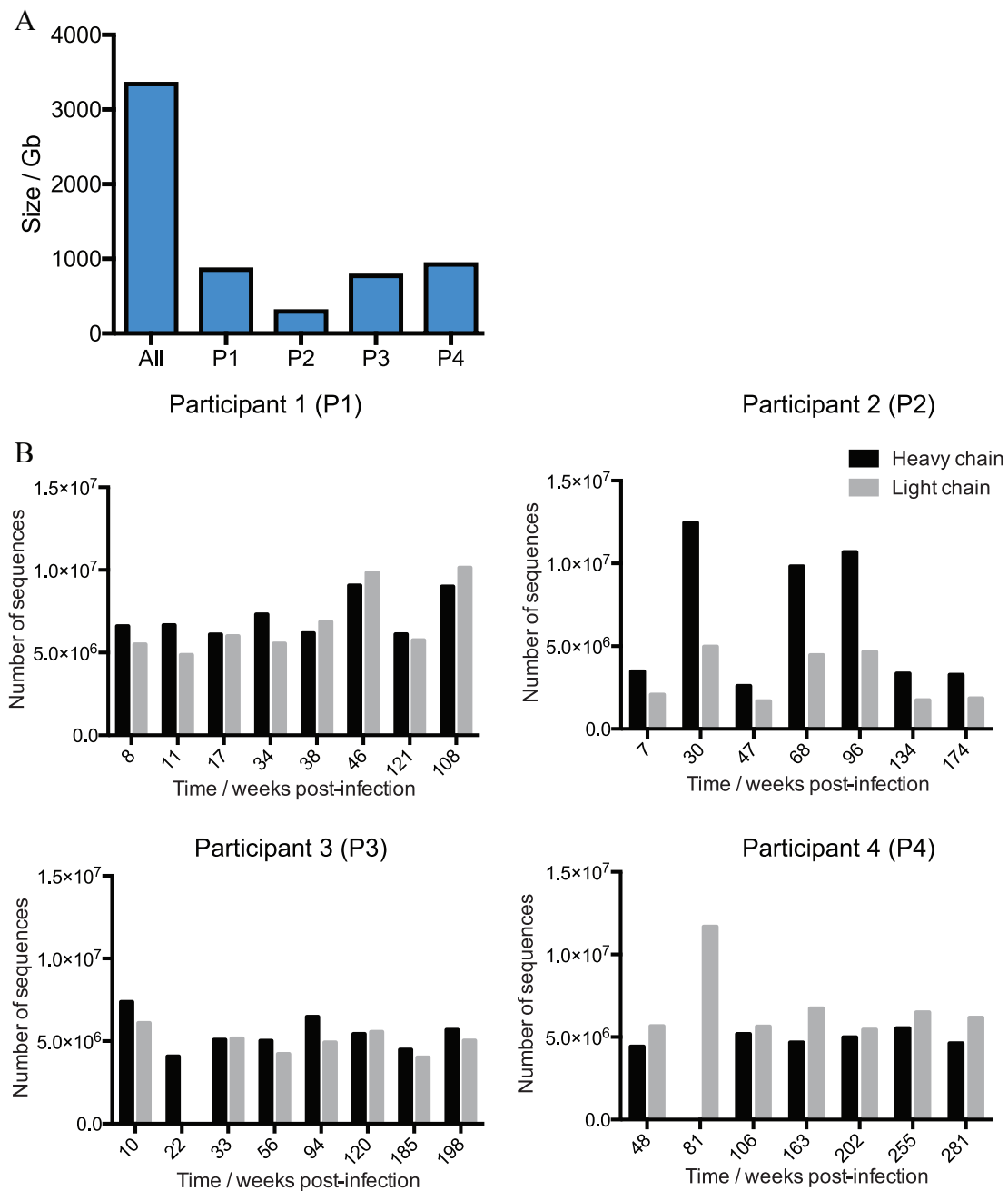


Figure 3. Cluster performance with large datasets of antibody repertoire data. (A) Total amount of data from the SONAR analyses, followed by the breakdown per participant for donors. (B) Number of antibody sequences analyzed using the bioinformatics cluster per donor and per time point. Heavy and light chain antibody sequences data are shown in black and gray, respectively.

programs on the cluster, in that it automatically updates the other nodes, whereas Ansible has to be run to update any changes on the other nodes. We used Ansible because it is simple and easy to configure compared with the other two as our cluster is small, with less than 3 nodes.

A centralized cluster for running tasks makes it easy to standardize processes and install updates on programs run by users. In addition, it enables good management of memory and storage resources as well as data security as users do not have to carry huge quantities of data around. Access to the cluster only requires use of the command line and an understanding of Unix systems. If a Windows operating system is preferred,

installation of a program such as Putty will provide a command line to execute Unix commands.

We applied the cluster to the analysis of the HIV envelope glycoproteins evolution within individuals¹⁹ and to studying the development of antibody responses to HIV infection.³⁰ These examples demonstrated how usage of this cluster has helped overcome hurdles in data processing and analysis and generated valuable insights into HIV vaccine design. This computational cluster was built using ordinary desktops, and the memory was upgraded to the maximum limits of the nodes. It will not perform well with analyses that require memory beyond the limit of the infrastructure, and more sophisticated

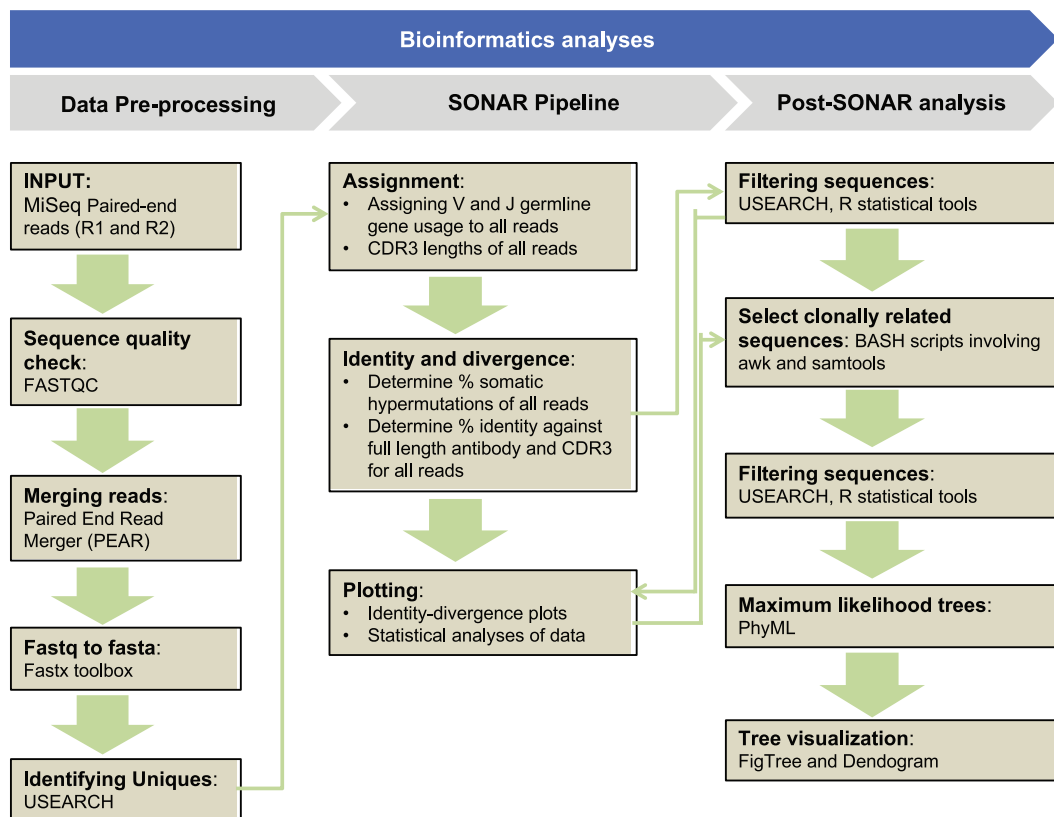


Figure 4. NGS analysis flowchart using the bioinformatics programs on the cluster. The antibody repertoire analysis involves first organizing the data from the sequencing facility (data pre-processing), SONAR analysis, and post SONAR analysis that involves selection of clonally related sequences. The arrows show the flow of data through the various stages, and in some instances, data can go back and forth in certain stages (shown by the thin arrows).

systems may be required. Future work would look into implementing parallel computing to break down large memory-demanding tasks into manageable jobs that can run within the limits of the available infrastructure.

Conclusion

In this article, we have demonstrated the building and implementation of a low-cost cluster for analyzing large NGS data and performing computationally intensive studies of intra-host evolution of HIV. The establishment of this low-cost cluster demonstrates how researchers from low-income settings can solve global challenges using relatively inexpensive resources. Such an approach of using low-cost technologies and recycling/repurposing equipment to tackle complex scientific problems is highly relevant to Africa and has broader implications in advancing creativity, research, and bringing about home-grown solutions to the challenges facing the continent.

Author Contributions

BMM conceived, designed and tested the high-performance computing cluster. RR and LD installed and maintained the cluster. PvH contributed to configuration of the cluster. LM, CS and PLM provided supervision to BMM. BMM, CS and PLM wrote the manuscript. All authors reviewed and approved the manuscript.

ORCID iDs

Batsirai M Mabvakure  <https://orcid.org/0000-0002-2760-5443>

Peter Van Heusden  <https://orcid.org/0000-0001-6553-5274>

Cathrine Scheepers  <https://orcid.org/0000-0002-1683-0282>

Penny L Moore  <https://orcid.org/0000-0001-8719-4028>

REFERENCES

- DePristo MA, Banks E, Poplin R, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet.* 2011;43:491-498. doi:10.1038/ng.806.
- Beerenwinkel N, Gunthard HF, Roth V, Metzner KJ. Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Front Microbiol.* 2012;3:329. doi:10.3389/fmicb.2012.00329.
- Li Y, Chen L. Big biological data: challenges and opportunities. *Genom Proteom Bioinform.* 2014;12:187-189. doi:10.1016/j.gpb.2014.10.001.
- Shoko R, Manasa J, Maphosa M, et al. Strategies and opportunities for promoting bioinformatics in Zimbabwe. *PLoS Comput Biol.* 2018;14:e1006480. doi:10.1371/journal.pcbi.1006480.
- Oxford Nanopore T. Minion. <https://nanoporetech.com/products/minion>. 2018. Accessed August 15, 2018.
- Kulkarni P, Frommolt P. Challenges in the setup of large-scale next-generation sequencing analysis workflows. *Comput Struct Biotechnol J.* 2017;15:471-477. doi:10.1016/j.csbj.2017.10.001.
- Alonso A, Lasseigne BN, Williams K, et al. aRNApipe: a balanced, efficient and distributed pipeline for processing RNA-seq data in high performance computing environments. *Bioinformatics.* 2017;33:1727-1729. doi:10.1093/bioinformatics/btx023.
- Schmidt B, Hildebrandt A. Next-generation sequencing: big data meets high performance computing. *Drug Discov Today.* 2017;22:712-717. doi:10.1016/j.drudis.2017.01.014.

9. Jerne NK. The natural-selection theory of antibody formation. *Proc Natl Acad Sci U S A*. 1955;41:849-857. doi:10.1073/pnas.41.11.849.
10. Tonegawa S. Somatic generation of antibody diversity. *Nature*. 1983;302:575-581. doi:10.1038/302575a0.
11. Glanville J, Zhai W, Berka J, et al. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proc Natl Acad Sci U S A*. 2009;106:20216-20221. doi:10.1073/pnas.0909775106.
12. Wiley SR, Raman VS, Desbien A, et al. Targeting TLRs expands the antibody repertoire in response to a malaria vaccine. *Sci Transl Med*. 2011;3:93ra69. doi:10.1126/scitranslmed.3002135.
13. Gaschen B, Taylor J, Yusim K, et al. Diversity considerations in HIV-1 vaccine selection. *Science*. 2002;296:2354-2360. doi:10.1126/science.1070441.
14. Fischer W, Ganusov VV, Giorgi EE, et al. Transmission of single HIV-1 genomes and dynamics of early immune escape revealed by ultra-deep sequencing. *PLoS ONE*. 2010;5:e12303. doi:10.1371/journal.pone.0012303.
15. Gruning B, Dale R, Sjodin A, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods*. 2018;15:475-476. doi:10.1038/s41592-018-0046-7.
16. van Loggerenberg F, Mlisana K, Williamson C, et al. Establishing a cohort at high risk of HIV infection in South Africa: challenges and experiences of the CAPRISA 002 acute infection study. *PLoS ONE*. 2008;3:e1954. doi:10.1371/journal.pone.0001954.
17. Salazar-Gonzalez JF, Bailes E, Pham KT, et al. Deciphering human immunodeficiency virus type 1 transmission and early envelope diversification by single-genome amplification and sequencing. *J Virol*. 2008;82:3952-3970. doi:10.1128/JVI.02660-07.
18. Doria-Rose NA, Schramm CA, Gorman J, et al. Developmental pathway for potent V1V2-directed HIV-neutralizing antibodies. *Nature*. 2014;509:55-62. doi:10.1038/nature13036.
19. Mabvakure BM, Lambson BE, Ramdayal K, et al. Positive selection at key residues in the HIV envelope distinguishes broad and strain-specific plasma neutralizing antibodies. *J Virol*. 2018;93:e01685-18. doi:10.1128/JVI.01685-18.
20. Mabvakure BM, Lambson BE, Ramdayal K, et al. Evidence for both intermittent and persistent compartmentalization of HIV-1 in the female genital tract. *J Virol*. 2019;93:e00311-19. doi:10.1128/JVI.00311-19.
21. Drummond AJ, Rambaut A. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol*. 2007;7:214. doi:10.1186/1471-2148-7-214.
22. Rambaut A, Drummond A. FigTree v1. 3.1. 2009. <http://tree.bio.ed.ac.uk/software/figtree/>
23. Bielejec F, Baele G, Vrancken B, Suchard MA, Rambaut A, Lemey P. Spred3: interactive visualization of spatiotemporal history and trait evolutionary processes. *Mol Biol Evol*. 2016;33:2167-2169. doi:10.1093/molbev/msw082.
24. Schramm CA, Sheng Z, Zhang Z, Mascola JR, Kwong PD, Shapiro L. SONAR: a high-throughput pipeline for inferring antibody ontogenies from longitudinal sequencing of B cell transcripts. *Front Immunol*. 2016;7:372. doi:10.3389/fimmu.2016.00372.
25. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. 2004;32:1792-1797. doi:10.1093/nar/gkh340.
26. Sievers F, Wilm A, Dineen D, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol Syst Biol*. 2014;7:539. doi:10.1038/msb.2011.75.
27. Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool. *J Mol Biol*. 1990;215:403-410.
28. Drummond AJ, Suchard MA, Xie D, Rambaut A. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Mol Biol Evol*. 2012;29:1969-1973. doi:10.1093/molbev/mss075.
29. Olsen GJ, Matsuda H, Hagstrom R, Overbeek R. fastDNAm1: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Comput Appl Biosci*. 1994;10:41-48.
30. van Eeden C, Wibmer CK, Scheepers C, et al. V2-directed vaccine-like antibodies from HIV-1 infection identify an additional K169-binding light chain motif with broad ADCC activity. *Cell Rep*. 2018;25:3123-3135.e6. doi:10.1016/j.celrep.2018.11.058.