High-Performance Computing of BEAST/BEAGLE in Bayesian Phylogenetics using SDumont Hybrid Resources

Kary Ocaña[1], Micaella Coelho[1], Guilherme Freire[1,2], Carla Osthoff[1]

[1]National Laboratory of Scientific Computing (LNCC)
Zip Code 25.651-075 – Petrópolis, RJ – Brazil

[2]Faculty of Technological Education of the State of Rio de Janeiro (FAETERJ)
Petrópolis, RJ – Brazil.

{karyann,micaella,freire,osthoff}@lncc.br

Abstract. *Bayesian phylogenetic algorithms are computationally intensive. BEAST 1.10 inferences made use of the BEAGLE 3 high-performance library for efficient likelihood computations. The strategy allows phylogenetic inference and dating in current knowledge for SARS-CoV-2 transmission. Follow-up simulations on hybrid resources of Santos Dumont supercomputer using four phylogenomic data sets, we characterize the scaling performance behavior of BEAST 1.10. Our results provide insight into the species tree and MCMC chain length estimation, identifying preferable requirements to improve the use of high-performance computing resources. Ongoing steps involve analyzes of SARS-CoV-2 using BEAST 1.8 in multi-GPUs.*

1. Introduction

Bayesian inference for phylogenetic analyses is computationally challenging due to the intensive nature of the likelihood calculations required to analyze the increasing size of molecular sequence character states under models of evolution. BEAST[1] [Suchard et al. 2018] package version 1.10 is a popular Bayesian implementation to assess the feasibility of conducting phylogenomic analyses of Coronavirus[2] epidemic outbreaks, reported in late 2019 in China. The virus phylogeny, evolutionary rates, times of the most recent common ancestor (tMRCA), and demographic growth are co-estimated using a Markov Chain Monte Carlo (MCMC) method in BEAST. BEAGLE 3 [Ayres et al. 2019] high-performance library is coupled to make efficient the fine-scale parallelization of phylogenetic likelihood calculations.

The need to test hundreds of replicates across many conditions can conduct larger analyses using BEAST. The increasing availability of phylogenomic sequence data motivates further improvements to the computational efficiency of fully Bayesian inference for analysis of hundreds or even thousands of sites across tens or hundreds of species. These improvements need to scale efficiently on many-core systems such as cluster supercomputers, as such systems offer vastly greater computing power than any desktop workstation. In the following sections, we describe the choices of parameters, models, and experimental simulation conditions for our computational experiments. We

[1] Bayesian Evolutionary Analysis by Sampling Trees
[2] Coronaviruses are RNA viruses of the family Coronaviridae including MERS (MERS-CoV) and SARS (SARS-CoV)

evaluate the scaling behavior of BEAST 1.10 in the Brazilian Santos Dumont[3] (SDumont) supercomputer and its statistical accuracy relative to several parameters to be considered to guarantee the efficient use of resources.

2. Related Work

New adaptations to accelerate MCMC calculations in GPU become more popular. BEAGLE 3 [Ayres et al. 2019] improves the use of phylogenetic software packages in clusters and multi-GPUs. It presents new parallel implementations to increased performance for challenging data sets, improved scalability, and better usability. New OpenCL and CPU-threads were added to the library to allow the effective utilization of a wider range of modern hardware. APIs support concurrent computation of independent partial likelihood arrays to increase the performance of nucleotide-model analyses with greater flexibility of data partitioning.

Viral phylodynamics [Volz et al. 2003] studies how epidemiological, immunological, and evolutionary processes act and potentially interact to shape viral phylogenies. Viral phylogenies represent the evolutionary impact of transmission dynamics on viral genetic variation selection, which are used to investigate important epidemiological, immunological, and evolutionary processes. Reconstructing pathogen viral phylodynamics from genetic data, as Coronavirus, is an emerging and open challenge. BEAST 1.10 Bayesian phylogenetic inference package becomes the preferred method [Giovanetti et al. 2020].

To manage increasing data flow in a Bayesian phylogenetic framework, [Gill et al. 2020] efficiently update the posterior distribution with newly available genetic data. This augmentation creates informed starting values and reuses optimally tuned transition kernels for posterior exploration of growing data sets, reducing the time necessary to converge to target posterior distributions. The Bayesian Skygrid model [Hill and Baele 2019] is a nonparametric coalescent model that estimates the effective population size over time used to infer past population dynamics over time from heterochronous molecular sequence data. MCMC approaches were proposed using an adaptive multivariate transition kernel to estimate in parallel a large number of parameters, split across partitioned data, by exploiting multi-core processing [Baele et al. 2017], [Bielejec et al. 2014]. Implementations enable the estimation of multipartite parameters more efficiently than standard approaches that typically use a mixture of univariate transition kernels. They are available in BEAST 1.10.

3. High-Performance Computing using BEAST through BEAGLE

BEAST [Suchard et al. 2018] software package has become a primary tool for Bayesian phylogenetic and phylodynamic inferences from genetic sequence data. BEAST unifies molecular phylogenetic reconstruction with complex discrete and continuous trait evolution, divergence-time dating, and coalescent demographic models in an efficient statistical inference engine using MCMC integration. In statistics, MCMC methods comprise a class of algorithms for sampling from a probability distribution. BEAST 1.10 presents a series of advances with a particular focus on delivering accurate and

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

informative insights for infectious disease research through the integration of diverse data sources, including phenotypic and epidemiological information, with molecular evolutionary models. The combination of BEAST 1.10 with BEAGLE 3 [Ayres et al. 2019] allows to parallelize multiple data partitions across a single high-performance device (i.e. GPU) to use the full capacity of these devices, reducing the computational overheads. BEAST 1.10 can yield a sizeable increase in effectively independent posterior samples per unit-time over previous software versions.

BEAGLE 3 [Ayres et al. 2019] is a high-performance library that can perform core calculations at the heart of most Bayesian and likelihood phylogenetic packages. It can make use of highly-parallel processors such as GPUs. Running BEAST 1.10 from the command line allows selecting the resource by typing `beast -beagle_info` shows a list of available resources that BEAGLE 3 has detected in SDumont: the CPU (always be available) and two NVidia Tesla K40t GPU card. BEAGLE 3 offers additional parallel computing advances and combines CUDA, OpenCL, and native CPU-threading implementations in a single codebase to address a wider-range of hardware resources. Additionally, it also benefits from GPU acceleration used to compute large trees and partitioned data sets. These advances serve as an important step in combining the capabilities of increasingly parallel hardware with the demands of progressively more sophisticated phylogenetic analyses.

BEAUti generates XML data files using alignments with default parameters for site models and Yule tree priors. BEAST 1.10 uses the flags `-overwrite -beagle_instances`, SSE runs, and BEAGLE 3 to start runs. The MCMC chain length was initially setting to 100,000 `<run id="mcmc" spec="MCMC" chainLength="100000">`. 20,000,000 `<run id="mcmc" spec="MCMC" chainLength="20000000">` was performed to make long enough extra likelihood calculations for debugging purposes with little effect on the outcome (Burn-In) and to eliminate JIT compiler differences. The computational resources were exclusively allocated with no other jobs running at the same time. The parameter `chainLength` specifies the number of steps the MCMC chain will make before finishing. This number depends on the size of the data set, the complexity of the model, and the precision of the answer required. The Burn-In is intended to give the Markov Chain time to reach its equilibrium distribution, where it is approaching the sampling distribution from its starting point and is usually discarded.

4. Experimental Results

This section presents the computational evaluation of BEAST 1.10 and BEAGLE 3. We have deployed bioinformatics applications and dependencies on the top of the SDumont. Software packages in the BEAST 1 family: BEAST[4] 1.10, BEAUti[5], TreeAnnotator[6]. Programs distributed independently used by the BEAST toolkit: FigTree[7] 1.4.4, and Tracer[8] 1.7.1. The BEAGLE 3[9] library for HPC was also deployed in SDumont.

[4] https://beast.community/beast. To take a control file and performs analysis
[5] https://beast.community/beauti. To import data, design analysis, and generate the control file
[6] https://beast.community/treeannotator. To produce a summary tree from BEAST outputs
[7] https://beast.community/figtree. To produce graphical outputs for viewing trees
[8] https://beast.community/tracer. To produce graphical outputs for diagnosing BEAST problems
[9] http://beast.community/beagle

4.1. SDumont Environmental Setup

SDumont has an installed processing capacity of 1.1 Petaflops per second with 18,424 CPU multi-cores, distributed over 758 computational nodes interconnected with an InfiniBand FDR/HDR network that provides high throughput and low latency for process communication and file system access. A parallel Lustre file system is integrated into the InfiniBand network with a gross storage capacity of 1.7 PBytes. SDumont contains types of devices with many-core architecture GPU and many integrated cores (MIC). GPU computing nodes contain 2 CPU Intel Xeon E5-2695v2 Ivy Bridge (12c @2,4GHz) and 64Gb RAM and GPU Nvidia K40. MESCA is a differentiated node with 240 cores and a large capacity shared memory architecture of 6 Tb. There is a special node for Artificial Intelligence applications of 8 NVIDIA Tesla V100-16Gb GPUs with NV link, totaling 40,960 CUDA-core and 5,120 Tensor-core. With its expanded capacity in late 2019 from 1.1 petaflops (2015) to 5.1 Petaflops per second with 34,688 CPU multi-cores distributed in 1,132 computational nodes, SDumont is on the world's Top500 list.

4.2. Experiment Setup

Four phylogenomic data sets were included in experiments: Yellow fever virus, Dengue virus, and two examples from the BEAST 1.10 benchmark directory. Table 1 presents the main features of data sets and parameters presented in the XML file used by BEAST 1.10. The last two columns refer to considerations (also suggested by the BEAST team) of using computational resources based on the type of data sets and evolutionary study.

Table 1. Genomic data features and XML file setting

Data set	Taxa number*	Alignment sites**	Alignment partition	Data & suggested resources[10]	Evolutionary study type considerations
DENV[11] - Genome	997	10,188	10	Large data; many taxa (hundreds); GPU	Rates of evolution and phylogenetic relationships for time-stamped sequences
Bench1[12] - Gene	1441	98	1	Small data; many taxa (hundreds); multi-threads	Benchmark for time-measured phylogenies
Bench2[13] - Genome	62	10,869	1	Large data; few taxa (dozens); GPU	Benchmark for time-measured phylogenies
YFV[14] - Gene	71	654	1	Medium data; few taxa (dozens); multi-threads	Phylogenetic relationships from partitioned sequences

* Taxon (pl. taxa). A taxonomic group of organisms. In this code, taxa may be organisms or species
** Site. Individual discrete position, normally a single nucleotide (or amino acid)

[10] http://www.phylo.org/index.php/tools/beast_how_fast. BEAST on BEAGLE Benchmarks
[11] https://github.com/beagle-dev/beagle-lib/tree/master/benchmarks/v3-app-note
[12] https://github.com/beast-dev/beast-mcmc/tree/master/examples/Benchmarks
[13] https://github.com/beast-dev/beast-mcmc/tree/master/examples/Benchmarks
[14] https://beast.community/rates_and_dates

Yellow fever virus remains the cause of severe morbidity/mortality in South America and Africa. YFV data set (rM/E gene region) [Bryant et al. 2007] was sampled from 22 countries over 76 years (until 2009) to estimate the rate of molecular evolution, date of tMRCA, and phylogenetic relationships with appropriate measures of statistical support. Dengue virus (DENV) complex (DENV-1 to DENV-4 serotypes) is the cause of the most common and important arthropod-borne viral disease of humans. DENV is predominantly transmitted by mosquitoes *Aedes aegypti* and *A. albopictus*, facilitating heavy viral transmission in densely populated tropical and subtropical regions. DENV data set [Ayres et al. 2019] is composed of 997 genomes spanning the global dengue diversity and 6,869 unique site patterns across 10 gene-based subsets. The data set of Benchmark 1 (Bench1) is composed of 1,441 taxa of a human alignment with 987 unique site patterns. The data set of Benchmark 2 (Bench2) is composed of 62 taxa of a mammal alignment with 5,565 unique site patterns.

4.3. Performance Results

BEAST can do many kinds of analyses, with Tree Likelihood calculations typically dominating the computational time of MCMC runs. To see the impact of the way BEAST handles in CPUs and GPUs, we performed analyses with and without data partition data sets. Since many analyses use different data types, we end up with four data sets (Table 1). We demonstrate the scaling behavior of BEAST 1.10 in SDumont, remarking: (i) BEAST 1.10 parallel execution is considerably improved using large and partitioned genomic data and (ii) supercomputer hybrid architecture enables to execute parallel jobs of BEAST 1.10 (multi-GPU is only available in BEAST 1.8).

Figure 1 presents the performance scalability as the total execution time (TET) and speedup of BEAST 1.10 setting `chainLength="100000"` in SDumont's MESCA nodes (until 240 threads) using DENV data set. Figure 2 presents TET of BEAST 1.10 setting `chainLength="100000"` and `chainLength="20000000"` in SDumont's GPUs using four data sets: DENV, Bench1, Bench2, YFV. All tests have been executed three times and we present the average bandwidth obtained over all three runs.

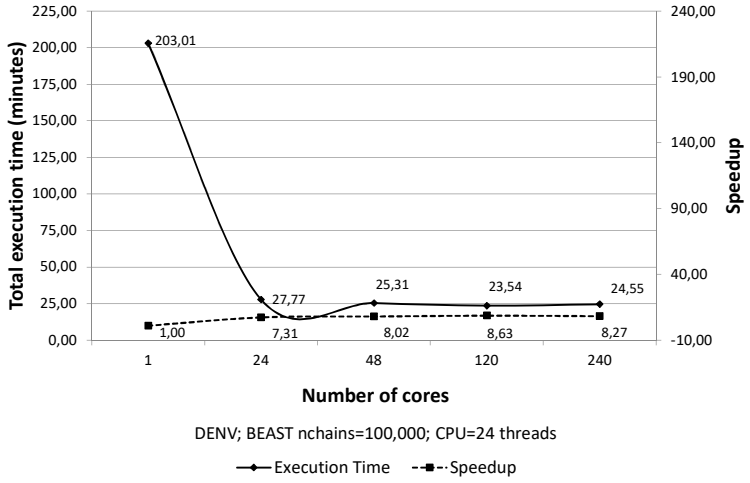

Figure 1. BEAST 1.10 TET and speedup for DENV data set in MESCA nodes

Figure 1 presents BEAST 1.10 performance behavior on a single processor machine (one core) to analyze the local optimization before scaling up the number of

threads to 240 in MESCA nodes. Analyses used the DENV largest input data set with 10 gene partitions and BEAST 1.10 with `chainLength="100000",` as suggested by [Ayres et al. 2019]. We observed that TET (in minutes) decreases from 203.01 (using one core) to 27.77 (using 24 threads), which means performance improvements of up to 86.32%. With an increasing number of threads, the difference in run time in minutes decreases, but over 24 threads, BEAST 1.10 no more outperforms. However, it can turn out that the data sets are too small for more than 24 threads.

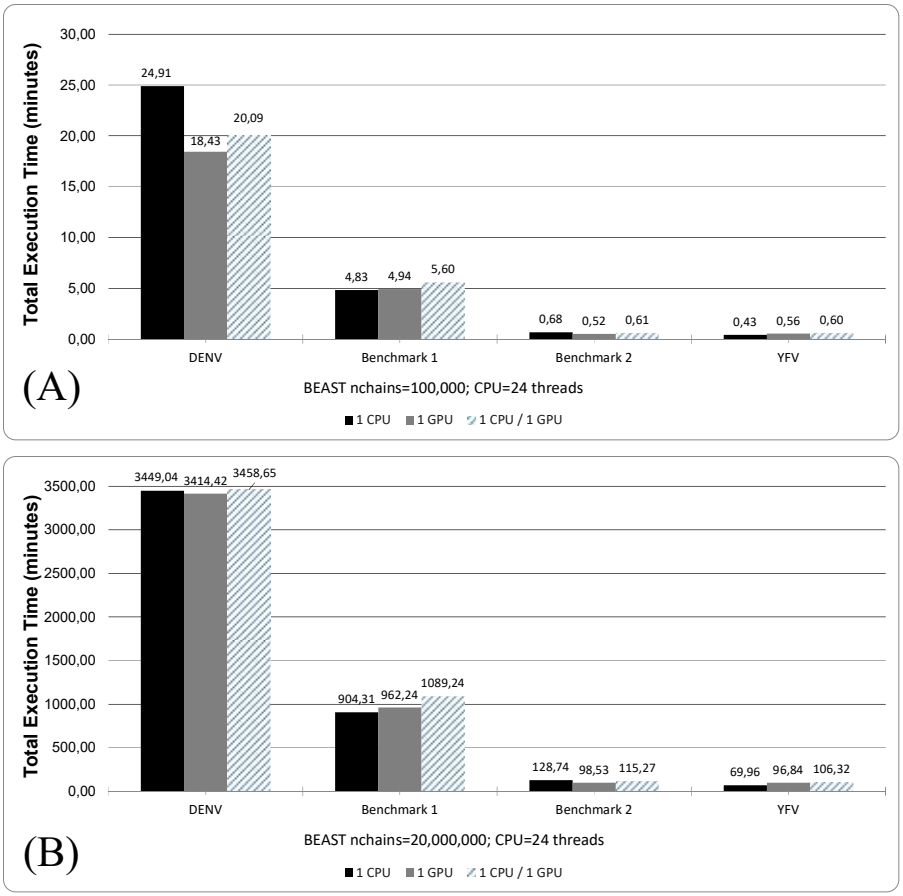

Figure 2. BEAST 1.10 TET for DENV, Bench1, Bench2, YFV in CPUs and GPUs. (A) `chainLength="100000"`; (B) `chainLength="20000000"`

To evaluate the behavior of performance gain according to the number of processing units, we used a speedup metric. An ideal speedup reduces the sequential time dividing this time by the number of processing units used. The speedup value was defined to evaluate performance gains of parallel computers and is impacted by serial portions of the code and communication between processors. The execution of BEAST 1.10 using 24 threads led to a speedup of 7.31. Even though there was always a gain by adding more threads, from 24 up to 240 threads, the speedup presented minimal degradation. This result indicates that providing more threads for execution may not bring the expected benefit, particularly if small data size files are involved. Note that in this study, we focus on virus data sets, using the largest genomes of eukaryotes or mammals must be performed to reinforce this behavior.

Figure 2 presents BEAST 1.10 over the four data sets DENV, Bench1, Bench2, and YFV. Data set presents very similar performance behavior executing BEAST 1.10

with `chainLength="100000"` in Figure 2 (A) and `chainLength="20000000"` in Figure 2 (B). The MCMC runs for 2 million steps make them long enough that the slightly different ways extra likelihood calculations are done at the start for debugging purposes have little effect on the outcome.

The effective sample size (ESS) rates for all logged statistics were greater than 200 and convergence was also confirmed graphically, demonstrating that BEAST can be applied to real phylogenetic data sets with many taxa. ESS is the number of independent samples that the trace is equivalent to, it is calculated as the chain length (excluding the Burn-In) divided by the auto-correlation time (ACT). ACT is the average number of states in the MCMC chain that two samples have to be separated by for them to be uncorrelated. The chain length specifies the number of steps the MCMC chain will make before finishing (i.e. the number of accepted proposals). It depends on the size of the dataset, the complexity of the model, and the precision of the answer required.

The partitioned DENV data set is the largest and benefits better from GPUs. Bench1, Bench2, and YFV have a single partition. Bench2 shows some benefit using GPUs, but Bench1 and YFV have too few unique sites per partition to benefit from using GPUs. CPU/GPU performs worse than only CPU or GPU for all data sets. Results using BEAST 1.10 using `chainLength="100000"` and `chainLength="20000000"`, respectively: **DENV** – GPU outperforms CPU in 6.48 and 34.62 minutes; **Bench2** – GPU outperforms CPU in 0.16 and 30.21 minutes; **Bench1** – CPU outperforms GPU in 0.13 and 57.93 minutes; **YFV** – CPU outperforms GPU in 0.11 and 26.88 minutes.

Our results suggest that the type of computational resource should offer better benefits depending on genomic data features and BEAST parameters, as `chainLength`. We agree with [Ayres et al. 2019], which test BEAST 1.6.1 using the BEAGLE framework on the Trestles at San Diego Supercomputer Center (SDSC). They used the benchmarks provided with the BEAST release to determine how to run BEAST with the BEAGLE library efficiently, concluding that Bench2 shows some benefits from using GPUs, but Bench1 had too few unique sites per partition to benefit from using GPUs.

5. Conclusion

We analyze the computational behavior of executing BEAST 1.10 and BEAGLE3 on CPU and GPU resources to provide a more complete usability experience to users at performing their phylogenomic data sets in high-performance resources. Our results using statistical predictions of BEAST based on performance, scaling, and usability of BEAGLE agree with [Ayres et al. 2019]. Novel evolutionary models of spreading and diversification of diseases are based on Bayesian time-scaled analysis implemented in BEAST 1.10, as presented by the actual pandemic COVID-19 [Giovanetti et al. 2020] reports. This is the importance to better understand the benefits of using high-performance computing resources to obtain evolutionary data in a feasible time, reducing as possible computational requirements.

Ongoing works cover several aims: (1) performance and scalability using MESCA nodes with BEAST 1.10 and 1.8 using `chainLength="20000000"`; (2) analyses of BEAST 1.8 using the 8 NVIDIA Tesla V100-16Gb GPUs with NV link of the updated SDumont (2019); (3) use of multi-thread CPU and multi-GPU using BEAST 1.10 and 1.8 and to analyze available SARS-CoV-2 data; (4) impact of the evolutionary

impacts using BEAST 2 over BEAST 1 to analyze SARS-CoV-2 data, it is suggested that BEAST 2 is almost always slightly faster than BEAST 1; and (5) cursory checks of ESS and MCMC for BEAST 1 and 2 to determine the run converged.

6. Acknowledgments

We thank Dr. Andre Soares for very helpful scientific suggestions about phylogenetics with BEAST. We are particularly grateful to the Santos Dumont Supercomputer Center at LNCC for substantial technological support. We thank the anonymous reviewers for valuable critiques. The financial support was provided by Brazilian grants CNPq/Universal (429328/2016-8) and FAPERJ/JCNE (232985/2017-03).

7. References

Ayres, D. L.; Cummings, M. P.; Baele, G.; Darling, A. E.; Lewis, P. O.; Swofford D. L.; Huelsenbeck, J. P.; Lemey, P.; Rambaut, A.; Suchard, M. A. (2019). "BEAGLE 3: Improved Performance, Scaling, and Usability for a High-Performance Computing Library for Statistical Phylogenetics". *Systematic Biology*, Volume 68, Issue 6, Pages 1052–1061.

Baele, G.; Lemey, P.; Rambaut, A.; and Suchard, M. A. (2017). "Adaptive MCMC in Bayesian phylogenetics: An application to analyzing partitioned data in BEAST". *Bioinformatics*, Volume 33, Issue 12, Pages 1798–1805.

Bielejec, F.; Lemey, P.; Carvalho, L. M.; Baele, G.; Rambaut, A.; and Suchard, M. A. (2014). "πBUSS: a parallel BEAST/BEAGLE utility for sequence simulation under complex evolutionary scenarios". *BMC Bioinformatics*, Volume 15, Issue 133.

Bryant, J. E.; Holmes, E. C.; and Barrett, A. D. T. (2007). "Out of Africa: A molecular perspective on the introduction of yellow fever virus into the Americas". *PLoS Pathogens*, Volume 3, Issue 5, Pages 668–673.

Gill, M. S.; Lemey, P.; Suchard, M. A.; Rambaut, A.; and Baele G. (2020). "Online Bayesian Phylodynamic Inference in BEAST with Application to Epidemic Reconstruction". *Molecular Biology and Evolution*, Volume 37, Issue 6, Pages 1832–1842.

Giovanetti, M.; Benvenuto, D.; Angeletti, S.; and Ciccozzi, M. (2020). "The first two cases of 2019-nCoV in Italy: Where they come from?" *Journal of Medical Virology*, Volume 92, Issue 5, Pages 518–521.

Hill, V. and Baele, G. (2019). "Bayesian estimation of past population dynamics in BEAST 1.10 using the Skygrid coalescent model". *Molecular Biology and Evolution*, Volume 36, Issue 11, Pages 2620-2628.

Suchard, M. A.; Lemey P.; Baele G.; Ayres D. L.; Drummond A. J.; and Rambaut A. (2018). "Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10". *Virus Evolution*, Volume 4, Issue 1, vey016.

Volz, E.M.; Koelle, K; and Bedford, T. (2013). "Viral Phylodynamics". *PLoS Computational Biology*, Volume 9, Issue 3, e1002947.