



HAL
open science

BEAGLE 3 High-performance Computational Library for Phylogenetic Inference

Daniel Ayres, Philippe Lemey, Guy Baele, Marc A Suchard

► **To cite this version:**

Daniel Ayres, Philippe Lemey, Guy Baele, Marc A Suchard. BEAGLE 3 High-performance Computational Library for Phylogenetic Inference. Scornavacca, Celine; Delsuc, Frédéric; Galtier, Nicolas. Phylogenetics in the Genomic Era, No commercial publisher | Authors open access book, pp.5.4:1–5.4:9, 2020. hal-02536457

HAL Id: hal-02536457

<https://hal.science/hal-02536457>

Submitted on 10 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

Chapter 5.4 BEAGLE 3 High-performance Computational Library for Phylogenetic Inference

Daniel L. Ayres

Center for Bioinformatics and Computational Biology, University of Maryland, USA
ayres@umd.edu

Philippe Lemey

Department of Microbiology, Immunology and Transplantation, Rega Institute, KU Leuven – University of Leuven, Leuven, Belgium

Guy Baele

Department of Microbiology, Immunology and Transplantation, Rega Institute, KU Leuven – University of Leuven, Leuven, Belgium

Marc A. Suchard

Department of Biomathematics, Department of Biostatistics, Department of Human Genetics, University of California, Los Angeles, CA 90095, USA

Abstract

Maximum-likelihood and Bayesian inference approaches to statistical phylogenetics require repeatedly computing of the observed sequence data likelihood. As increasingly cheaper sequencing technology provides phylogenomic studies with larger data set sizes, there stands a critical need to efficiently evaluate this likelihood function in order for phylogenetic computations to complete in reasonable time. The adoption of powerful computing architectures, in the form of multi-core central processing units and many-core graphics cards dedicated to scientific computing, offers unprecedented opportunity to perform massively parallel computation in many research fields, including likelihood evaluation in phylogenetics. In this chapter, we provide insight into the inner workings of BEAGLE, a high-performance likelihood-calculation platform for use on multi-core and many-core computer systems (ubiquitous nowadays in standard desktop computers and laptops) and available in several phylogenetic inference applications to improve computational performance.

How to cite: Daniel L. Ayres, Philippe Lemey, Guy Baele, and Marc A. Suchard (2020). BEAGLE 3 High-performance Computational Library for Phylogenetic Inference. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter No. 5.4, pp. 5.4:1–5.4:9. No commercial publisher | Authors open access book. The book is freely available at <https://hal.inria.fr/PGE>.

1 BEAGLE 3 high-performance computational library


BEAGLE defines a uniform application programming interface (API) and includes a collection of efficient implementations for evaluating the phylogenetic likelihood function under a wide range of evolutionary models, on multi-core central processing units (CPUs) and, importantly, many-core graphics processing units (GPUs). The BEAGLE library can be installed as a shared resource, to be used by any software aimed at phylogenetic reconstruction that supports the library. This approach allows developers of phylogenetic software packages to share in optimizations of the core calculations and for any program that uses BEAGLE to benefit from improvements to the library. For researchers, this centralization provides a single installation to take advantage of new hardware and parallelization techniques.




© Daniel L. Ayres, Philippe Lemey, Guy Baele and Marc A. Suchard.
Licensed under Creative Commons License CC-BY-NC-ND 4.0.

Phylogenetics in the genomic era.

Editors: Celine Scornavacca, Frédéric Delsuc and Nicolas Galtier; chapter No. 5.4; pp. 5.4:1–5.4:9

 A book completely handled by researchers.

 No publisher has been paid.

5.4:2 BEAGLE 3 library

BEAGLE is typically associated with BEAST, and in fact, recent versions of BEAST require the presence of the BEAGLE library in order to perform phylogenetic and/or phylodynamic inference. This is also the case for the analyses performed in the chapter on “Efficiently analysing large viral data sets in computational phylogenomics,” that makes extensive use of the BEAGLE (Ayres et al., 2019) library. Many of its features also are used within MrBayes (Ronquist et al., 2012) and BEAST 2.5 (Bouckaert et al., 2019). Further, support for maximum-likelihood inference has been available since BEAGLE’s inception, with significant speedups observed in packages such as GARLI (Zwickl, 2006) and PhyML (Guindon et al., 2010), and with work currently underway for PAUP* (Swofford, 2003).

BEAGLE version 3 (Ayres et al., 2019), the latest release of the library, enables analyses with data partitions or with few site patterns to benefit from significant performance increases on GPUs. It also adds new CPU-threaded and GPU software implementations, allowing more effective utilization of a wide range of modern parallel processors. BEAGLE 3 is free, open-source software licensed under the Lesser GPL and available for Windows, Mac and Linux from <https://github.com/beagle-dev/beagle-lib/releases>.

1.1 Parallel Computation with BEAGLE

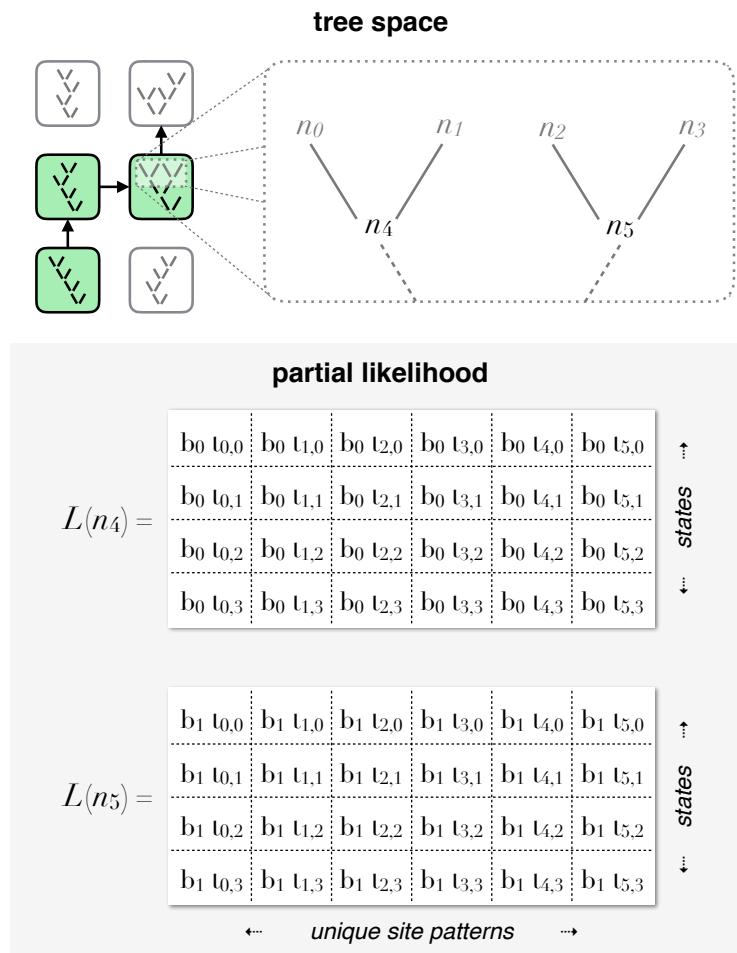
Advances in computer hardware, specifically in parallel architectures, such as multi-core CPUs, CPU intrinsics (e.g., SSE, AVX) and many-core GPUs, have created opportunities to speed up computationally intensive methods. The structure of the likelihood calculation, involving large numbers of positions and multiple states, as well as other characteristics, makes it a very appealing computational fit to these modern parallel processors, especially to GPUs. Recognizing that different independent calculations are possible in computing phylogenetic likelihoods, as well as that different computing hardware architectures have different strengths with respect to parallel computation, BEAGLE implements concurrent computation in a variety of ways. These can be roughly categorized into three broad levels based on granularity: fine-, medium- and course-grained parallelism.

1.1.1 Fine-Grained Parallelism

BEAGLE exploits GPUs via fine-grained parallelization of functions necessary for computing the likelihood on a phylogenetic tree. Phylogenetic inference programs typically explore tree space in a sequential manner (Figure 1, *tree space*) or with a small number of sampling chains, thus offering a low upper limit for coarse-grained parallelization. In contrast, the crucial computation of partial likelihood arrays at each node of a proposed tree presents an excellent opportunity for fine-grained data parallelism, for which GPUs are especially suited. The use of many lightweight execution threads incurs very low overhead on GPUs and the presence of large numbers of positions and multiple states enables efficient parallelism at this level (Figure 1, *partial likelihood*).

Furthermore, BEAGLE uses GPUs to parallelize other functions necessary for computing the overall tree likelihood, thus minimizing data transfers between the CPU and GPU. These additional functions include those necessary for computing branch transition probabilities, for integrating root and edge likelihoods, and for summing site likelihoods.

BEAGLE also provides SSE and OpenCL implementations for exploiting fine-grained parallelism on CPUs that vectorize likelihood calculations across characters and character states. These solutions, however, offer only a modest performance benefit as CPU vectorization intrinsics are of limited width (128 bits are available with SSE and up to 512 bits with AVX vectorization). Additionally, CPU architectures have lower memory bandwidth than



■ **Figure 1** Diagrammatic example of the tree sampling process and medium and fine-grained parallel computation of phylogenetic partial likelihoods using BEAGLE on GPUs for a nucleotide-model problem with 5 taxa, 5 site patterns. Each entry in a partial likelihood array L is assigned to a separate GPU thread t , and each array is assigned to a separate GPU execution block b . In this simplified example, 48 GPU threads are created to enable parallel evaluation of each entry of the partial likelihood arrays $L(n_4)$ and $L(n_5)$.

5.4:4 BEAGLE 3 library

GPUs and we have found this to be a limiting factor when it comes to fine-grained parallel computation of phylogenetic likelihoods.

1.1.2 Medium-Grained Parallelism

In order to calculate the overall likelihood of a proposed tree, phylogenetic inference programs perform a tree traversal, evaluating a partial likelihood array at each node. With BEAGLE, the evaluation of these multi-dimensional arrays is offloaded to the library. Further, when these partial likelihood arrays are independent from one another, they may also be evaluated in parallel to one another, with BEAGLE assigning the calculation of each array to separate execution blocks on the GPU (Figure 1, *partial likelihood*).

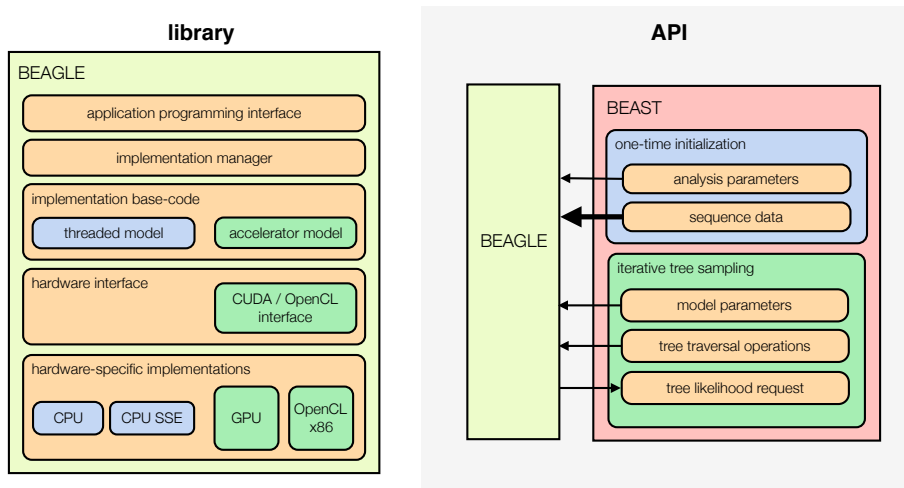
With BEAGLE version 3, partitioned analyses also benefit from multi-core CPUs and GPUs by parallelizing the computation of multiple data subsets (Ayres and Cummings, 2017b,a). This capability suits the trend of phylogenomic data sets that are often heavily partitioned in order to better model the underlying evolutionary processes.

1.1.3 Coarse-Grained Parallelism

Phylogenetic inference programs that implement multiple Markov chain Monte Carlo chains or independent runs can invoke multiple BEAGLE library instances, one for each chain or run. For effective parallelism, this is more efficient on multiple hardware resources (e.g., multiple GPUs) and for each library instance to be assigned to a separate resource.

1.2 Library and API Design

1.2.1 Library



■ **Figure 2** Layer diagrams depicting the BEAGLE library organization, and illustration of API use. Arrows indicate direction and relative size of data transfers between the client program and library.

The general structure of the BEAGLE library can be conceptualized as a set of layers (Figure 2, *library*), the uppermost of which is the application programming interface (API). Underlying this API is an implementation management layer, which loads the available

implementations, makes them available to the client program, and passes API commands to the selected implementation.

The design of BEAGLE allows for new implementations to be developed without the need to alter the core library code or how client programs interface with the library. This architecture also includes a plugin system that allows implementation-specific code (via shared libraries) to be loaded at runtime when the required dependencies are present. Consequently, new frameworks and hardware platforms can more easily be made available to programs that use the library, and ultimately to users performing phylogenetic analyses.

The implementations in BEAGLE version 3 derive from two general models. One is a threaded CPU implementation model that does not directly use external frameworks. Under this model, there is a parallel CPU implementation, and one with added SSE intrinsics that uses vector processing extensions present in many CPUs to further parallelize computation across character state values.

The other implementation model involves an explicit parallel accelerator programming model, and uses the CUDA and the OpenCL external computing frameworks to exploit parallel hardware (Ayres and Cummings, 2017b). It implements fine-grained and medium-grained parallelism for evaluating likelihoods under arbitrary molecular evolutionary models, thus being able to harness large numbers of processing cores to efficiently perform calculations (Suchard and Rambaut, 2009; Ayres et al., 2019).

At the lowest implementation level in BEAGLE, functions that impart a crucial effect on performance are differentiated for each hardware type. This allows for distinctly optimized parallel implementations that are shown in Figure 2, one for NVIDIA and OpenCL-compatible GPUs and one for OpenCL-compatible x86 parallel resources such as multicore CPUs with SIMD-extensions.

1.2.2 Application Programming Interface

The BEAGLE API was designed to increase performance via parallelization while reducing data transfer and memory copy overhead to an external hardware accelerator device (e.g., GPU). Client programs, such as BEAST (Suchard et al., 2018), use the API to offload the evaluation of tree likelihoods to the BEAGLE library (Figure 2, *API*). API functions can be subdivided into two categories: those which are only executed once per inference run and those which are repeatedly called as part of an iterative sampling process. For the one-time initialization process, client programs use the API to indicate analysis parameters such as tree size and sequence length, as well as specifying the type of evolutionary model and hardware resource(s) to be used. This allows BEAGLE to allocate the appropriate number and size of data buffers on device memory. Also at this initialization stage, the sequence data are specified and transferred to device memory. This costly memory operation is only performed once, thus minimizing its impact.

During the iterative tree sampling procedure, client programs use the API to specify changes to the evolutionary model and instruct a series of partial likelihood operations that traverse the proposed tree in order to find its overall likelihood. BEAGLE efficiently computes these operations and makes the overall tree likelihood as well as per-site likelihoods available via another API call.

2 BEAGLE in practice

2.1 Performance

Peak performance with BEAGLE is achieved when using a high-end GPU, with the relative gain over using a CPU depending on model type and problem size as more demanding analyses allow for better utilization of GPU cores. Figure 3 shows speedups relative to single-core CPU code when using BEAGLE on multiple CPU cores and on an NVIDIA Tesla P100 GPU for calculating the likelihood function under a nucleotide model, with increasing unique site pattern counts. Computing the likelihood function typically accounts for over 90% of the total execution time for phylogenetic inference programs and the relationship between speedups and problem size observed here primarily matches what would be observed for a full analysis.

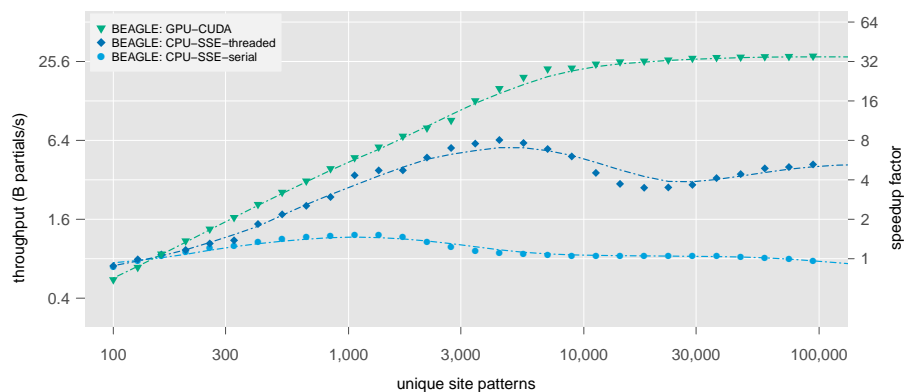


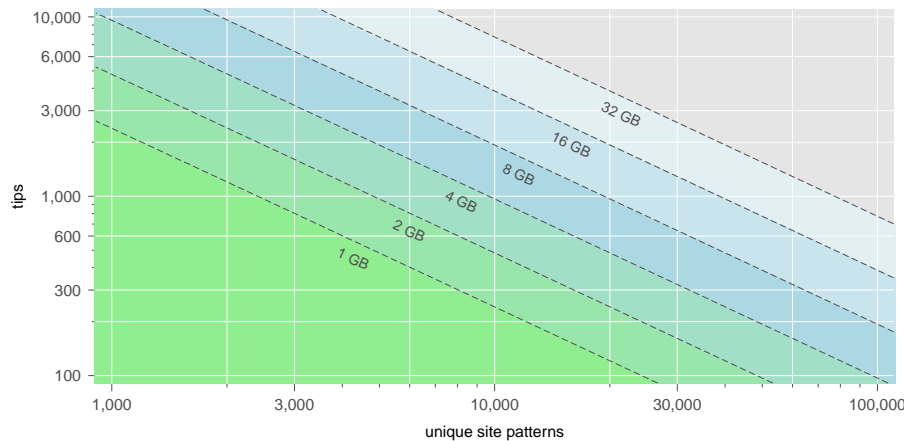
Figure 3 Absolute (throughput in billions of partial likelihood calculations per second) and relative (fold-speedup relative to the slowest performance observed for the BEAGLE library CPU with SSE at any number of unique site patterns) performance for native implementations of the BEAGLE library version 3 on an Intel Xeon E5-2690v4 CPU and NVIDIA Tesla P100 GPU. The data are simulated nucleotide sequences for a tree with 128 tips.

Using a nucleotide model, relative GPU performance over the CPU strongly scales with the number of site patterns. For very small numbers of patterns the GPU exhibits poor performance due to greater execution overhead relative to overall problem size. GPU performance improves quickly as the number of unique site patterns is increased and by 10,000 patterns it is closer to a saturation point, continuing to increase but with diminishing returns. At 100,000 nucleotide patterns the GPU is approximately 64× faster than the serial CPU implementation.

For partitions with higher state counts, such as those found in discrete traits models used in phylodynamic analyses, GPU speedup may be achieved with a single character. This is due to the better parallelization opportunity afforded by the increased number of states that can be encoded by the character. The higher state count of such data compared to nucleotide data increases the ratio of computation to data transfer, resulting in increased GPU performance for each character.

2.2 Memory usage

When assessing the suitability of GPU acceleration via BEAGLE for a phylogenetic analysis, it is also important to consider if the GPU has sufficient on-board memory for the analysis to be performed. GPUs typically have less memory than what is available to CPUs and the high transfer cost of moving data from CPU to GPU memory prevents direct use of CPU memory for GPU acceleration.



■ **Figure 4** Log-log contour plot depicting BEAGLE-GPU memory usage for BEAST nucleotide model analyses with four-rate categories and double precision floating-point arithmetic, over a range of problem sizes in terms of number of tips and of unique site patterns. The amount of memory depicted as values below to dashed isolines convey the upper boundary for the memory size indicated. Memory requirements shown here assume an unpartitioned dataset. Partitioned analyses incur a small amount of memory overhead, typically less than 100 MB.

Figure 4 shows how much memory is required for problems of different sizes when running nucleotide model partitions in BEAST (Suchard et al., 2018) with BEAGLE GPU acceleration. Note that when multiple GPUs are available, BEAST can split the data into separate BEAGLE instances, one for each GPU. Thus each GPU will only require as much memory as necessary for the data subset assigned to it. Typical PC-gaming GPUs have 8 GB of memory or less, while GPUs dedicated to high performance computing, such as the NVIDIA Tesla series, currently have as much as 32 GB of memory.

For partitions with a discrete trait character, the memory required depends on the number of states the character can assume, in addition to the size of the tree (i.e., the number of tips). This memory requirement will typically be significantly less than that of the nucleotide data. As an example, for a tree with 1,000 tips, a discrete trait character partition with 100 possible states will use approximately 0.5 GB of GPU memory.

2.2.1 Hardware

Highly parallel computing technologies such as GPUs have overtaken traditional CPUs in peak performance potential and continue to advance at a faster pace. Additionally, the memory bandwidth available to the processor is especially relevant to data-intensive computations, such as the evaluation of nucleotide model likelihoods. In this measure as well, high-end GPUs significantly outperform equivalently positioned CPUs.

BEAGLE was designed to take advantage of this trend of increasingly advanced GPUs and uses runtime compilation methods to optimize code for which-ever generation of hardware is being used. For the analyses in the “Efficiently analysing large viral data sets in computational phylogenomics” chapter, we have used an NVIDIA Tesla P100 GPU, with 3584 CUDA cores and 16 GB of memory. Its peak figures for memory bandwidth and computational performance are of 720 GB/s and 4.7 trillion floating-point operations per seconds (TFLOPS). These figures are nearly an order of magnitude higher than those for a modern, high-end multi-core CPU. At the time of writing, the most powerful GPU for scientific computing is NVIDIA’s Tesla V100, which comes equipped with 5120 CUDA cores and 32 GB of memory, with a memory bandwidth of 900 GB/s and a computational performance of up to 7.8 TFLOPS.

3 Discussion

BEAGLE is a high-performance computational library that offers substantial performance gains in phylogenetic and phylodynamic inference. Now at version 3, BEAGLE is fully integrated with BEAST 1.10.5 (Suchard et al., 2018) and MrBayes 3.2.7 (Ronquist et al., 2012), making use of the latest advances such as increased parallelism for nucleotide-model analyses on GPUs. We note that another high-performance library, known as the Phylogenetic Likelihood Library (Flouri et al., 2015), has been developed and integrated in two phylogenetic software packages: DPPDiv (Heath et al., 2011) and IQ-TREE (Nguyen et al., 2015). While benchmark tests (Flouri et al., 2015; Ayres et al., 2019) identify the strengths of both libraries in different scenarios, it is apparent that support for these libraries remains rather limited at the time of writing. We expect that with ever-growing data set sizes, both libraries will be increasingly adopted over time allowing parameter estimation for complex evolutionary models.

While offering substantial performance gains for statistical phylogenetics, the use of such high-performance libraries is not a panacea to enable complex model combinations on increasingly large data sets. Employing these libraries should be coupled with highly-efficient parameter estimation strategies, which have started to find their way into Bayesian phylogenetic and phylodynamic inference. To enable parallel estimation of a potentially large collection of continuous parameters, adaptive MCMC is able to exploit multi-core processing architectures to improve MCMC integration efficiency (Baele et al., 2017). Bayesian phylogenetics has also started adopting Hamiltonian Monte Carlo to improve inference efficiency of branch-specific evolutionary rates, by means of fast gradient evaluations (Ji et al., 2019). These efficient transition kernels have only recently seen their first implementations in phylogenetics and phylodynamics research, but offer promising avenues for further performance improvements.

References

- Ayres, D. L. and Cummings, M. P. (2017a). Configuring concurrent computation of phylogenetic partial likelihoods: Accelerating analyses using the BEAGLE library. In Ibrahim, S., Choo, K., Yan, Z., and Pedrycz, W., editors, *Algorithms and Architectures for Parallel Processing. ICA3PP 2017, Helsinki, Finland*, volume 10393 of *Lect. Notes Comput. Sc.*, pages 533–547.
- Ayres, D. L. and Cummings, M. P. (2017b). Heterogeneous hardware support in BEAGLE, a high-performance computing library for statistical phylogenetics. In *46th International*

- Conference on Parallel Processing Workshops (ICPPW 2017)*, pages 23–32, Bristol, United Kingdom.
- Ayres, D. L., Cummings, M. P., Baele, G., Darling, A. E., Lewis, P. O., Swofford, D. L., Huelsenbeck, J. P., Lemey, P., Rambaut, A., and Suchard, M. A. (2019). BEAGLE 3: Improved performance, scaling, and usability for a high-performance computing library for statistical phylogenetics. *Syst. Biol.*, 68(6):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*, 33(12):1798–1805.
- Bouckaert, R., Vaughan, T. G., Barido-Sottani, J., Duchêne, S., Fourment, M., Gavryushkina, A., Heled, J., Jones, G., Kühnert, D., De Maio, N., Matschiner, M., Mendes, F. K., Müller, N. F., Ogilvie, H. A., du Plessis, L., Poppinga, A., Rambaut, A., Rasmussen, D., Siveroni, I., Suchard, M. A., Wu, C.-H., Xie, D., Zhang, C., Stadler, T., and Drummond, A. J. (2019). Beast 2.5: An advanced software platform for bayesian evolutionary analysis. *PLOS Computational Biology*, 15(4):1–28.
- Flouri, T., Izquierdo-Carrasco, F., Darriba, D., Aberer, A., Nguyen, L.-T., Minh, B., Haeseler, A. V., and Stamatakis, A. (2015). The Phylogenetic Likelihood Library. *Syst. Biol.*, 2(1):356–362.
- Guindon, S., Dufayard, J.-F. F., Lefort, V., Anisimova, M., Hordijk, W., and Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.*, 59(3).
- Heath, T. A., Holder, M. T., and Huelsenbeck, J. P. (2011). A Dirichlet Process Prior for Estimating Lineage-Specific Substitution Rates. *Molecular Biology and Evolution*, 29(3):939–955.
- Ji, X., Zhang, Z., Holbrook, A., Nishimura, A., Baele, G., Rambaut, A., Lemey, P., and Suchard, M. A. (2019). Gradients do grow on trees: a linear-time $O(N)$ -dimensional gradient for statistical phylogenetics. <https://arxiv.org/pdf/1905.12146.pdf>.
- Nguyen, L.-T., Schmidt, H. A., von Haeseler, A., and Minh, B. Q. (2015). IQ-TREE: A Fast and Effective Stochastic Algorithm for Estimating Maximum-Likelihood Phylogenies. *Molecular Biology and Evolution*, 32(1):268–274.
- Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). Mrbayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, 61(3):539–542.
- Suchard, M., Lemey, P., Baele, G., Ayres, D., Drummond, A., and Rambaut, A. (2018). Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evolution*, 4(1):vey016.
- Suchard, M. A. and Rambaut, A. (2009). Many-core algorithms for statistical phylogenetics. *Bioinformatics*, 25:1370–1376.
- Swofford, D. L. (2003). *Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4*. Sinauer Associates, Sunderland, Massachusetts.
- Zwickl, D. J. (2006). *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion*. PhD thesis, The University of Texas at Austin.