Supplement to “ParGenes: a massively parallel tool for model selection and tree inference on thousands of genes”

Benoit Morel¹, Alexey M. Kozlov¹, Alexandros Stamatakis¹,²

¹Scientific Computing Group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
²Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany

Received on xx.xx.xxxx; revised on XXXXX; accepted on XXXXX

Associate Editor: xxxx xxx

1 EXPERIMENTAL SETUP

1.1 Datasets

We benchmarked ParGenes using two empirical gene family datasets.

The first one³ was initially used in Anselmetti et al. (2018) and comes from the Ensembl database (Zerbino et al., 2018). It contains 8,880 gene families from 15 mammalian species. The second dataset⁴ was obtained from the VectorBase database (Emrich et al., 2015) and contains 12,000 gene families of 15 Anopheles gambiae species, the primary mosquito vector responsible for the transmission of malaria in most of sub-Saharan Africa.

Scheduling independent maximum likelihood (ML) tree searches on these gene family MSAs is challenging, because of their varying dimensions, both, in terms of the number of sites, and the number of taxa (see Fig. 1). Note that, we count MSA lengths in terms of distinct MSA site patterns, as identical site patterns can be and are compressed by all phylogenetic inference tools in a pre-processing step.

1.2 Hardware

We executed our benchmarks on our institutional cluster that is equipped with 224 nodes with Intel Haswell CPUs (E5-2630v3) running at 2.40GHz. Each node has 2 CPUs and each CPU has 8 physical cores. The nodes have 64GB RAM and are connected via an Infiniband interconnect.

1.3 Benchmarks

We executed two distinct benchmarking runs.

- FULL Benchmark: for each gene family (each MSA), execute model testing, then 20 ML tree searches (from both, random, and parsimony starting trees) and 100 bootstrap tree inferences, select the best ML tree on the original MSA, and compute the bootstrap support values on that tree.
- FAST Benchmark: run a single ML tree search per gene family (per MSA).

³ Available at https://github.com/YoannAnselmetti/ADseq-Anopheles-APBC2018/blob/master/data/FASTA/MSA/CDS/MUSCLE.tar.gz
⁴ Available at https://sco.h-its.org/exelixis/material/ensembl_8880_15.tar.gz

The FULL Benchmark covers the complete feature set of ParGenes and thus represents the realistic default use-case. The load balance of this benchmark is likely to be ‘good’, even using a naive scheduling strategy, as it generates a comparatively large number of independent jobs (e.g., more than one million RAxML-NG runs for the 8800 gene family MSAs from Ensembl). While the FAST benchmark might also corresponds to a realistic use case (e.g., rapid initial data exploration), it generates substantially less inference jobs. This benchmark is thus more relevant and interesting for assessing the quality of our load balance strategy.

2 IMPACT OF LOAD BALANCE STRATEGY

In this Section, we show that an appropriate scheduling strategy is required to attain ‘good’ load balance. To this end, we analyzed our empirical datasets using the FAST benchmark (one ML search per family/MSA, see Section 1.3) using distinct scheduling strategies.

In the first naive ParGenes implementation, we initially sorted the jobs by descending order of predicted execution time. Then, we dynamically assigned each job to the available cores. Rather unsurprisingly, we observed that, some per-MSA inference jobs require substantially longer execution times than the average job, and that they continue running when most of the other jobs have completed. This resulted in a high proportion of inactive cores and, as a consequence, in poor load balance (see Fig. 2a).

To overcome this issue, we changed our strategy to assigning one or more cores (depending on the MSA size) to each job. However, assigning too many cores to a job reduces its parallel efficiency, and might thus induce a slowdown. We therefore determined the optimal per-job core number. The corresponding function is implemented in RAxML-NG. As mentioned in the main text, the jobs that require the highest number of cores are executed first. As shown in Fig. 2b, this heuristic improves the load balance and reduces the overall execution time by almost a factor of 5 in the specific experiment.

However, we still observe some ‘tails’, that is, jobs that are still running when most others are done, even when the number of cores
assigned to an individual MSA inference job is high. We observed that these tail jobs always correspond to per-MSA inference with a large number of taxa. To alleviate the tails, we assign twice the number of cores than recommended by RAxML-NG, to the top 5% of the per-MSA jobs with the largest number of taxa (top 5% heuristic). This is done, to (i) reduce their execution times and (ii) to also have them scheduled earlier. On the one hand, these jobs will typically exhibit sub-optimal parallel efficiency. on the other hand, they only account for a small portion of the overall workload and the performance impact of tails is alleviated. As shown in Fig. 2c, the top 5% heuristic shortens the tails and decreases overall runtime (15% improvement in the specific experiment).

3 EXPERIMENTAL RESULTS FOR FAST BENCHMARK

We executed the FAST benchmark (one ML search per gene family, see Section 1.3) on the VectorBase and the Ensembl datasets, using between 32 and 1024 cores. Our results (Table 1) show that, the ParGenes parallelization and scheduling strategy scales well up to 512 cores for both datasets. Using 512 cores ParGenes inferred the 8,880 and 12,000 ML trees in 35 minutes and 3 minutes, respectively, while attaining high parallel efficiency.
Table 1. Benchmark FAST: execution times (time) and parallel efficiencies (efficiency) for both VectorBase (VB.) and Ensenbl (En.) datasets with different numbers of cores. The parallel efficiency is relative to the run with the lowest number of cores.

| Cores  | 32 | 64 | 128 | 256 | 512 | 1024 |
|--------|----|----|-----|-----|-----|------|
| VB. time | 2713s | 1327s | 662s | 339s | 190s | 143s |
| VB. efficiency | 1.0 | 1.0 | 1.0 | 0.89 | 0.60 |      |
| En. time | 33150s | 16660s | 8342s | 3899s | 2050s | 1330s |
| En. efficiency | 1.0 | 1.0 | 0.99 | 1.06 | 1.01 | 0.78 |

Table 2. Execution times and parallel efficiency for the benchmark FULL applied to the VectorBase dataset, with different number of cores. The parallel efficiency is relative to the run with the lowest number of cores.

| Cores  | 256 | 512 | 1024 |
|--------|-----|-----|------|
| Execution time | 33000s | 18000s | 10800s |
| Efficiency | 1.0 | 0.91 | 0.76 |

4 EXPERIMENTAL RESULTS FOR FULL BENCHMARK

In this Section, we describe the results of the FULL benchmark (see Section 1.3; model testing, ML search with multiple distinct starting trees, bootstrap replicates).

ParGenes executed the entire analysis on the Ensembl dataset in 25 hours using 1024 cores. It achieved an overall load balance above 99%, which indicates that this analysis is likely to scale with much more cores. Most of the time (more than 24 hours) is spent in computing the best ML trees and the bootstraps replicates. ModelTest-NG ran in 25 minutes. All the other intermediate steps took in total 5 minutes. We successfully restarted the computations after ParGenes stopped because of the 24-hours wall time limit, using our checkpoint mechanism.

The VectorBase gene families contains much fewer genes per family than the Ensembl dataset (see Fig. 1), and is thus faster to process. We executed benchmark FULL on this dataset with different number of cores to show that it scales correctly (Table 2).

ACKNOWLEDGEMENT

This work was financially supported by the Klaus Tschira Foundation and by DFG grant STA 860/4-2. The colored plots were inspired by Nyan Cat (Torres and saraj00n, 2011).

REFERENCES

Anselmetti, Y., Duchemin, W., Tannier, E., Chauve, C., and Bérard, S. (2018). Phylogenetic signal from rearrangements in 18 Anopheles species by joint scaffolding extant and ancestral genomes. *BMC Genomics*, 19(Suppl 2).

Emrich, S. J. *et al.* (2015). VectorBase: an updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. *PLoS One*, 10(12), e0154274.

Zerbino, D. R. *et al.* (2018). Ensembl 2018. *NAR*, 46(D1), D754–D761.