Supplementary information for `phyx` (Phylogenetic tools for Unix)

Author, Co-Author, Stephen A. Smith

Department of Ecology & Evolutionary Biology
University of Michigan, Ann Arbor, MI 48109, USA

1 Performance

We briefly describe below the performance of `phyx` relative to other existing tools.

1.1 Sequence cleaning

TODO: JFW Cleaning sequences to ensure a certain level of matrix occupancy has become common place in many phylogenomic pipelinesAgalma,Yang2014. Here we compare two programs GblocksGblocks and phyutility [4] to the sequence cleaning procedure of Phyx (pxclsq). The file sizes ranged from 10 sequences in the file (234Kb), to 100,000 sequences (2.3Gb) with all being 23,950 base pairs in length. We found that Phyx outperformed both Gblocks and Phyutility in all dataset sizes and for the largest dataset Phyutility was not able to clean the dataset due to a memory allocation error. The test was conducted on a laptop containing 4 processors and 16Gb of memory, with 14Gb of that memory being allocated for phyutility.

1.2 Conversion of proteins to codons

TODO: JFW Converting the alignment of proteins to their corresponding nucleotide file is useful in helping ensure accuracy in the nucleotide alignment. Here we tested files that were between 10 sequences of 801bp for the amino acid alignment(8.0kb) and 100,000 sequences of(77M). We found that in all situations Phyx performed was faster than PAL2NAL[5] and a major advantage is that the sequences are not required to be in the same order, thus helping to avoid potentially aligning a nucleotide sequence with something other than its corresponding amino acid alignment.

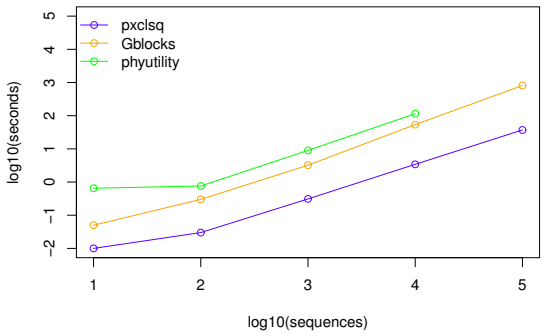

Figure 1: Comparison of alignment cleaning timings.

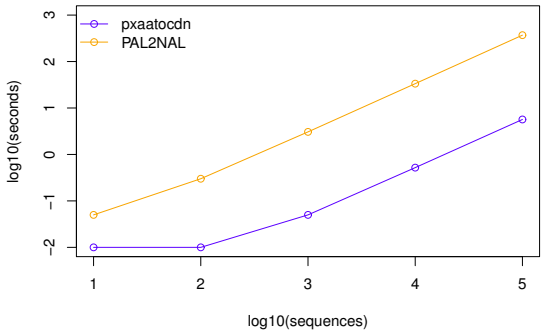

Figure 2: Comparison of timings to convert protein alignemtns to their corresponding codon alignments.

1.3 MCMC log concatenation and resampling

MCMC log files from Bayesian phylogenetic analyses have become common phylogenetic objects. Such analyses are typically replicated (to ensure convergence of the MCMC chains) and run for many millions of generations (to achieve adequate effective sample sizes), resulting in many several very large text files, each of which invariably involve a burnin phase (samples that are discarded before sumamrization). Prior to parameter summary, these log files are typically concatenated while removing the burnin phase and potentially resampling (thinning) the individual logs because of memory constraints. The `phyx` program `pxlog` carries out these operations on both tree and parameter logs. To assess the performance of `pxlog`, we compared it to two versions of `logcombiner`

from the BEAST package [2][1]. We ran phylogenetic in analyses in BEAST using the data from [3], a data set which consists of 798 taxa. Five replicates MCMC analyses were peformed, each running for 100 million generations and sampling trees every 5000 generations (for a total of 20000 trees sampled in each analysis). In preparation for tree summary, we discarded the first 25% of samples, and further thinned the chains to every 10th sample (for a total of 1500 post-burnin samples per analysis).

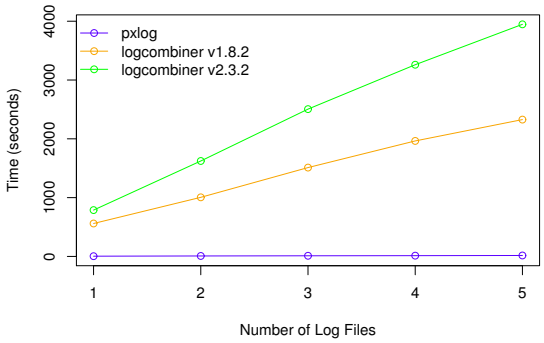

Figure 3: Comparison of MCMC log manipulation timings. Each log file is 2.6 GB and contains 20000 trees.

The timings for the log manipulations by the various programs are displayed in Figure 3. `pxlog` executed much faster than either version of `logcombiner` for any number of input files, taking only a few seconds compared to up to over an hour for the alternative tools. More revealing, however, was the memory usage of the various programs. `pxlog`, being stream-centric (and hence holding only a single tree in memory at any particular instant), consumed only 600 kb of RAM, despite the individual log files totalling 2.6 GB. `logcombiner` is a java-based tool to which we allocated 40 GB of RAM. `logcombiner` v1.8.2 was far more memory efficient than the newer version, consuming 2.4 GB of RAM for the full 5 file concatenation. `logcombiner` v2.3.2, on the other hand, consumed 32.6 GB of RAM while executing far more slowly.

2 Conclusions

We worked hard, and achieved very little.

References

[1] R. Bouckaert, J. Heled, D. Kühnert, T. Vaughan, C.-H. Wu, D. Xie, M. A. Suchard, A. Rambaut, and A. J. Drummond. Beast 2: A software platform

for bayesian evolutionary analysis. *PLoS Comput Biol*, 10(4):1–6, 04 2014.

[2] A. J. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7(1):1–8, 2007.

[3] S. Magallón, S. Gmez-Acevedo, L. L. Snchez-Reyes, and T. Hernández-Hernández. A metacalibrated time-tree documents the early rise of flowering plant phylogenetic diversity. *New Phytologist*, 207(2):437–453, 2015. 2014-18158.

[4] S. A. Smith and C. W. Dunn. Phyutility: a phyloinformatics tool for trees, alignments and molecular data. *Bioinformatics*, 24(5):715–716, 2008.

[5] M. Suyama, D. Torrents, and P. Bork. PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. *Nucleic Acids Research*, 34(suppl 2):W609–W612, 2006.