A peer-reviewed version of this preprint was published in PeerJ on 28 January 2016.

Borowiec ML. 2016. AMAS: a fast tool for alignment manipulation and computing of summary statistics. PeerJ 4:e1660 https://doi.org/10.7717/peerj.1660

AMAS: a fast tool for alignment manipulation and computing of summary statistics

Marek L Borowiec

The amount of data used in phylogenetics has grown explosively in the recent years and many phylogenies are inferred with hundreds or even thousands of loci and many taxa. These modern phylogenomic studies often entail separate analyses of each of the loci in addition to multiple analyses of subsets of genes or concatenated sequences. Computationally efficient tools for handling and computing properties of thousands of single-locus or large concatenated alignments are needed. Here I present AMAS (Alignment Manipulation And Summary), a tool that can be used either as a stand-alone command-line utility or as a Python package. AMAS works on amino acid and nucleotide alignments and combines capabilities of sequence manipulation with a function that calculates basic statistics. The manipulation functions include conversions among popular formats, concatenation, extracting sites and splitting according to a pre-defined partitioning scheme, and creation of replicate data sets. The statistics calculated include the number of taxa, alignment length, total count of matrix cells, overall number of undetermined characters, percent of missing data, AT and GC contents (for DNA alignments), count and proportion of variable sites, count and proportion of parsimony informative sites, and counts of all characters relevant for a nucleotide or amino acid alphabet. AMAS is particularly suitable for very large alignments with hundreds of taxa and thousands of loci. It performs better at concatenation and summarizing alignments than other popular tools. AMAS is a Python 3 program that relies solely on Python's core modules. AMAS source code and manual can be downloaded from http://github.com/marekborowiec/AMAS/

AMAS: a fast tool for alignment manipulation and computing of summary statistics

Marek L. Borowiec[1]

[1]**Department of Entomology and Nematology, University of California, Davis**

ABSTRACT

The amount of data used in phylogenetics has grown explosively in the recent years and many phylogenies are inferred with hundreds or even thousands of loci and many taxa. These modern phylogenomic studies often entail separate analyses of each of the loci in addition to multiple analyses of subsets of genes or concatenated sequences. Computationally efficient tools for handling and computing properties of thousands of single-locus or large concatenated alignments are needed. Here I present AMAS (Alignment Manipulation And Summary), a tool that can be used either as a stand-alone command-line utility or as a Python package. AMAS works on amino acid and nucleotide alignments and combines capabilities of sequence manipulation with a function that calculates basic statistics. The manipulation functions include conversions among popular formats, concatenation, extracting sites and splitting according to a pre-defined partitioning scheme, and creation of replicate data sets. The statistics calculated include the number of taxa, alignment length, total count of matrix cells, overall number of undetermined characters, percent of missing data, AT and GC contents (for DNA alignments), count and proportion of variable sites, count and proportion of parsimony informative sites, and counts of all characters relevant for a nucleotide or amino acid alphabet. AMAS is particularly suitable for very large alignments with hundreds of taxa and thousands of loci. It performs better at concatenation and summarizing alignments than other popular tools. AMAS is a Python 3 program that relies solely on Python's core modules. AMAS source code and manual can be downloaded from `http://github.com/marekborowiec/AMAS/`

Keywords: bioinformatics, phylogenomics, concatenation, alignment properties

INTRODUCTION

The amount of data used in modern phylogenetics has increased dramatically since the advent of next-generation sequencing (McCormack et al., 2013). Data sets composed of hundreds or thousands of loci are becoming commonplace. As a result of this, computer simulation studies also often require manipulation of thousands of alignments (Arenas, 2012). Efficient concatenation of alignments, for example, is important in phylogenetics and multiple concatenation procedures are needed in order to explore how various subsets of the data compare in inference (e.g. slow-evolving vs. fast-evolving loci (Sharma et al., 2014)). Alignment summary statistics are also useful in exploratory data analysis and can serve as a basis for filtering out "gappy" or fast-evolving data from downstream analyses (Borowiec et al., 2015). Modern phylogenetic analysis often requires a form of bioinformatics pipeline where output of one procedure is being redirected as input for another tool. Although a number of freely available tools for manipulating alignments and computing their basic statistics exist, some of the most popular ones are based on graphical user interfaces (e.g. Mesquite: Maddison and Maddison (2015)) and not appropriate for command-line or scripted pipeline analyses. Other command-line tools that have functionality partially overlapping with AMAS (FASconCAT-G: Kück and Longo (2014), Phyutility: Smith and Dunn (2008)) have a broader range of functions. More specifically, in addition to conversion, concatenation, and producing alignment summaries, FASconCAT-G allows the user to, among many other functions, write MrBayes blocks in NEXUS files or create consensus sequences. Phyutility also allows for interactions with the NCBI databases and a number of manipulations on phylogenetic trees. While AMAS currently lacks these capabilities, unlike the two other applications, it can be imported as a Python module to harness the potential of this programming language for further processing of its output. As demonstrated below, AMAS also outperforms these programs at concatenation of large data sets and computing of

alignment statistics. It is easy to install and use, requires only a standard distribution of Python 3.0 or newer, and is provided with a detailed instructions manual.

METHODS

To assess and compare the performance of the concatenation fuctions of AMAS, FASconCAT-G, and Phyutility, I used four recently published phylogenomic data sets: filtered DNA alignments of 8,295 exons of 52 vertebrates and 3,769 UCE loci of 49 bird taxa Jarvis et al. (2014) (available at `gigadb.org/dataset/101041`), DNA and amino acid alignments of 1,478 loci from 144 arthropod taxa from Misof et al. (2014) (`http://datadryad.org/resource/doi:10.5061/dryad.3c0f1`), and amino acid alignments of 5,214 exons from 19 wasp taxa from Johnson et al. (2013) (`datadryad.org/resource/doi:10.5061/dryad.jt440`).

FASconCAT-G v1.02 (Kück and Longo, 2014) was downloaded from `https://www.zfmk.de/en/research/research-centres-and-groups/fasconcat-g` and used according to the user manual provided along with the software download. Concatenation runs with FASconCAT-G were done in two modes: with the `-i` option that prevents the program from simultaneous calculation of alignment statistics for faster computing times, and with simultaneous writing of the statistics. Phyutility (Smith and Dunn, 2008) is available at `https://code.google.com/p/phyutility/`. It was used according to the manual. For a complete list of verbatim commands used for benchmarking see Table S1.

The performance was assessed on a desktop computer equipped with 12 Intel(R) Core(TM) i7-4930K CPUs at 3.40Ghz, 32GB of DDR3 RAM at 1600Hz, a generic 4TB 7200rpm hard drive, and a Ubuntu 12.04LTS operating system.

I used the standard Unix command `time -p` to evaluate execution times. Each command was run three times and best time recorded.

FUNCTIONALITY AND USAGE

The open source code of AMAS and its manual are available at `http://github.com/marekborowiec/AMAS/` or at Python's Package index at `https://pypi.python.org/pypi/amas/`. AMAS has been tested on a number of data sets to assure that all supported formats are correctly parsed. It is a project under active development and, following user suggestions, additional features are likely to be implemented in the future.

AMAS can be imported as a Python module or used as a command line application. The module interface has the advantage of much greater flexibility in processing input and output, while the command line interface can be used for quick tasks and remains accessible to users with little Python scripting experience.

AMAS as a Python package

AMAS uses the custom `MetaAlignment` class and its methods to handle multiple sequence alignments. All the major functions available on the command line are available through this interface. A discussion of available methods is available in the manual supplied with the package. I focus on the command line options when discussing AMAS capabilities below.

Command line interface

AMAS requires the user to supply information on the alignment: 1) the name of input file(s), 2) the input format, and 3) whether it contains nucleotide or amino acid sequences. The user also needs to specify at least one action to be performed by the program. These actions are explained below. AMAS uses Python's `argparse` module to handle user-provided arguments (or flags) and is agnostic of the order in which they are given. Correct specification of the input format is crucial for correct parsing of the files. While AMAS takes minimal measures to detect if the right format was given, there is a trade-off between computation time and automated detection of file formats.

Conversion among formats

AMAS allows easy conversion of formats for thousands of files in seconds. It is able to parse five of the most popular formats of multiple sequence alignments: FASTA, PHYLIP (both sequential and interleaved

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1355v1 | CC-BY 4.0 Open Access | rec: 8 Sep 2015, publ: 8 Sep 2015

formats are supported), and NEXUS (sequential and interleaved). Note that conversions to interleaved formats are much less efficient than conversions among sequential formats and may take several minutes for very large data sets.

An example of conversion from FASTA to NEXUS (sequential) is given below.

```
$ AMAS.py -d dna -f fasta -i *fas -v -u nexus
```

As mentioned above, the user is required to provide the information about the input by specifying the format, data type, and name of the file or files to be processed. The format flags `-f` or `--in-format` (long version) and `-u` or `--out-format` are used to specify the input and output formats, respectively. Supported are `fasta`, `phylip` ('relaxed' i.e. taxon names with >20 characters are permitted), `phylip-int`, and `nexus-int`.

Two data types are recognized (`-d` or `--data-type`): `aa` for amino acid alignments or `dna` for DNA sequences.

The name of the input file (`-i` or `--in-file`) can be a single file name or use shell's expansion to signify multiple files. The input in the example above `*fas` (assuming we are using a Unix shell) will mean all file names in the current directory terminating with the `fas` extension.

The action to convert among formats is specified above with `-v` (equivalent to `--convert`).

Concatenation

AMAS allows fast concatenation and the input and output files can be in any of the supported formats. It also creates a partitions file that records the coordinates of each locus in the concatenated alignment. AMAS is competitive relative to other popular software used for this purpose: see Performance section below.

Creating replicate datasets

AMAS allows the user to create concatenated alignments from a number of randomly chosen alignments that can be used for, for example, the phylogenetic jackknife. Assuming you have a large set of input files (e.g. thousands of single-locus alignments or arbitrary sets of sites created by splitting with AMAS; see below), you can create any number of replicate alignments, each concatenated from any number of files randomly chosen from the input set.

Extracting sites and splitting by partition

AMAS allows writing files from input alignments, given a list of sites. This can be used to split alignments by partition in matrices where the data originally supplied is only in the form of concatenated matrix but with information on partitioning schemes used. The output of this action is produced according to a site/partition file that should include unique names for each partition or file to be written and their coordinates in the alignment to be split. Codon positions are easily accessed through the standard notation where the range of coordinates is given, followed by a backslash and the integer three. For example:

```
AApos1&2 = 1-604\3, 2-605\3
AApos3 = 3-606\3
28SAutapoInDels=7583, 7584, 7587, 7593
```

The above partition file would produce three output files, one containing sites from positions one and two in the gene `AA`, one containing the third codon position sites from the same marker, and a third file containing four sites from a different region.

Computing alignment statistics

AMAS can calculate basic alignment properties that include the number of taxa, alignment length, total number of matrix cells, overall number of undetermined characters, percent of missing data, AT and GC contents (for DNA alignments), number and proportion of variable sites, number and proportion of parsimony informative sites, and counts of all characters relative to the matrix size.

Missing data includes any completely undetermined characters as well as gaps. For DNA alignments these characters include `X`, `N`, `O`, `-`, `?` and for amino acid alignments the characters `X`, `.`, `*`, `-`, `?`. The number of variable and parsimony-informative sites is calculated after removing missing and undetermined characters from each site. AT and GC content are calculated including the ambiguity codes `W` and `S`, respectively. AT/GC contents are also calculated only relative to each other and not to the overall contents of the matrix, i.e. excluding any missing data.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1355v1 | CC-BY 4.0 Open Access | rec: 8 Sep 2015, publ: 8 Sep 2015

PERFORMANCE

I assessed the performance of AMAS on four alignments ranging from, when concatenated, approximately 57 to 705 million matrix cells in total and occupying ca. 56 to 681 megabytes of hard drive space as FASTA files. The alignment length in total number of sites (amino acids or DNA bases), no. of matrix cells, percent of missing data, and proportion of parsimony-informative sites, as calculated by AMAS, are given in Table 1.

Table 1. Select statistics of benchmark alignments.

Alignment name	Data type	Length	Total cells	Missing percent	Prop. variable	Prop. parsimony inf.
Johnson et al. (2013)	amino acid	3,001,657	57,031,483	46.42	0.53	0.28
Misof et al. (2014)	amino acid	1,313,129	189,090,576	64.43	0.76	0.59
Jarvis et al. (2014) UCEs	nucleotide	9,251,694	453,333,006	19.46	0.64	0.44
Jarvis et al. (2014) exons	nucleotide	13,557,123	704,970,396	16.63	0.51	0.35

Conversion among formats

Conversion among formats is fast with AMAS. The largest data set of 8,295 alignment files from Jarvis et al. (2014) was converted among sequential formats in 15-20 seconds (see Table S1).

Concatenation

Concatenation is a function that can be performed by two other popular programs that are used for alignment manipulations in phylogenomic data sets: FASconCAT-G, a Perl program (Kück and Longo, 2014) and Phyutility, written in Java (Smith and Dunn, 2008). The former supports FASTA, PHYLIP, and CLUSTAL alignments as input, the latter only FASTA and NEXUS formats. Here I compare the performance of both to AMAS. FASconCAT-G worked on the amino acid alignments of Johnson et al. (2013) and Misof et al. (2014) data sets, concatenating the former in over 70 seconds and the latter in about 260 seconds. This program also allows simultaneous computation of alignment summaries during concatenation. This took about 8 and 23 minutes for the two datasets, respectively. FASconCAT-G crashed with an error when attempting to read in the nucleotide FASTA files of Jarvis et al. (2014) UCE and exon data sets. Phyutility was able to concatenate all of the four FASTA data sets, in times of about 10 minutes for the smallest data set of 5,214 amino acid loci of Johnson et al. (2013) to over 200 minutes for the largest data set of 8,295 exons from Jarvis et al. (2014). Phyutility allows concatenation only to the sequential NEXUS format. AMAS concatenation times ranged from about 2 seconds for the smallest data set of Johnson et al. (2013) to about 22 seconds for Jarvis et al. (2014) data, outperforming the other two programs by a factor of 30 or more. A comparison of times taken for concatenating the smallest data set of 5,214 amino acid loci from Johnson et al. (2013) is presented in Figure 1A. See Table S1 for additional comparisons.

Computing summary statistics

The times required for computing summaries are similar regardless of whether the data sets are processed as many single-locus alignments or a concatenated matrix and I present the results on concatenated alignments only (Figure 1B). These are also similar regardless of the input file format. See Table S1 for an exhaustive list of benchmarks. FASconCAT-G (Kück and Longo, 2014) also calculates alignment statistics but it is significantly slower than AMAS. The total time taken for FASconCAT-G for concatenation and summary computing on the smallest data set of Johnson et al. (2013) was >500 seconds, while AMAS concatenates this data set in 2 seconds and outputs summaries in another 30 seconds (Figure 1).

Splitting by partition

The process of splitting concatenated files is also very fast. AMAS wrote 5,214 files from partitions of the Johnson et al. (2013) concatenated matrix in about 5 seconds, and 8,295 files from the Jarvis et al. (2014) exon data set in about 30 seconds.

DISCUSSION

AMAS is a fast program, suitable for a variety of simple manipulations useful in phylogenetic inference. It is robust to various input data formats and outperforms other popular programs at concatenation and

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1355v1 | CC-BY 4.0 Open Access | rec: 8 Sep 2015, publ: 8 Sep 2015

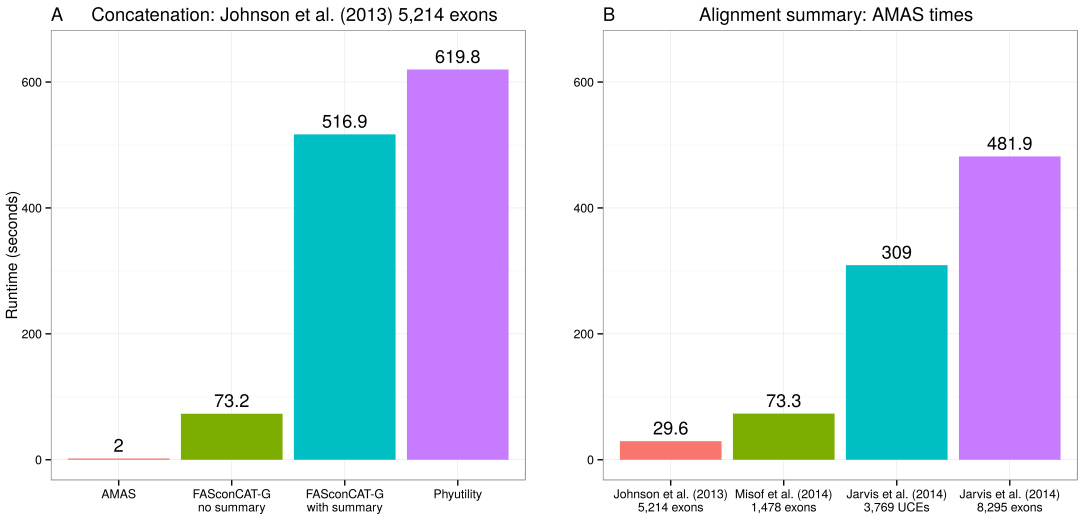

Figure 1. Performance
. A: Computing times for concatenation of the Johnson et al. (2013) data set composed of 5,214 separate alignments. FASconCAT-G was run in two modes: with and without simultaneous computation of alignment summaries. B: Computing times for AMAS writing alignment summaries on the four benchmark data sets.

computing alignment statistics. It is also potentially much more flexible because of its design as a Python package. Future improvements to AMAS are anticipated, including functions to manipulate the alignments on by taxon basis.

The continuing growth in the amount of sequence data used in phylogenetics requires that even faster similar tools be developed soon, taking advantage of compiled languages such as C++ or Julia. At present, however, AMAS offers a tool that is available to any potential user with access to a command line interface, allowing fast computing on some of the largest alignments published to date.

ACKNOWLEDGMENTS

I would like to thank Carlos Peña for help in the initial stages of this project and Brian Johnson for access to the test desktop computer. Phil Ward provided comments that helped to improve this manuscript.

REFERENCES

Arenas, M. (2012). Simulation of molecular data under diverse evolutionary scenarios. *PLoS Computational Biology*, 8:e1002495.

Borowiec, M. L., Lee, E. K., Chiu, J. C., and Plachetzki, D. C. (2015). Dissecting phylogenetic signal and accounting for bias in whole-genome data sets: a case study of the Metazoa. bioRxiv. Published online January 20, 2015. http://dx.doi. org/10.1101/013946.

Jarvis, E. D., Mirarab, S., Aberer, A. J., Li, B., Houde, P., Li, C., Ho, S. W., C., F. B., Nabholz, B., Howard, J. T., Suh, A., Weber, C. C., da Fonseca, R. R., Alfaro-Núñez, A., Narula, N., Liu, L., Burt, D., Ellegren, H., Edwards, S. V., Stamatakis, A., Mindell, D. P., Cracraft, J., Braun, E. L., Warnow, T. Jun, W., Gilbert, M. P., Zhang, G., and The Avian Phylogenomics Consortium (2014). Phylogenomic analyses data of the avian phylogenomics project. *GigaScience*, 4:4.

Johnson, B. R., Borowiec, M. L., Chiu, J. C., Lee, E. K., Atallah, J., and Ward, P. S. (2013). Phylogenomics resolves evolutionary relationships among ants, bees, and wasps. *Current Biology*, 23:2058–2062.

Kück, P. and Longo, G. C. (2014). FASconCAT-G: extensive functions for multiple sequence alignment preparations concerning phylogenetic studies. *Frontiers in Zoology*, 11:81.

Maddison, W. P. and Maddison, D. R. (2015). Mesquite: a modular system for evolutionary analysis http://mesquiteproject.org. Version 3.04.

McCormack, J. E., Hird, S. M., Zellmer, A. J., Carstens, B., and Brumfield, T. R. (2013). Applications of next-generation sequencing to phylogeography and phylogenetics. *Molecular Phylogenetics and Evolution*, 66:526–538.

Misof, B., Liu, S., Meusemann, K., Peters, R. S., Donath, A., Mayer, C., Frandsen, P. B., Ware, J., Flouri, T., Beutel, R. G., Niehuis, O., Petersen, M., Izquierdo-Carrasco, F., Wappler, T., Rust, J., Aberer, A. J., Aspöck, U., Aspöck, H., Bartel, D., Blanke, A., Berger, S., Böhm, A. Buckley, T. R., Calcott, B., Chen, J., Friedrich, F., Fukui, M., Fujita, M., Greve, C., Grobe, P. Gu, S., Huang, Y., Jermiin, L. S., Kawahara, A. Y., Krogmann, L., Kubiak, M., Lanfear, R., Letsch, H., Li, Y., Li, Z., Li, Z., Lu, H., Machida, R., Mashimo, Y., Kapli, P., McKenna, D. D., Meng, G., Nakagaki, Y., Navarrete-Heredia, J. L., Ott, M., Ou, Y., Pass, G., Podsiadlowski, L., Pohl, H., von Reumont, B. M., Schütte, K., Sekiya, K., Shimizu, S., Slipinski, A., Stamatakis, A., Song, W., Su, X., Szucsich, N. U., Tan, M., Tan, X., Tang, M., Tang, J., Timelthaler, G., Tomizuka, S., Trautwein, M., Tong, X., Uchifune, T., Walzl, M. G., Wiegmann, B. M., Wilbrandt, J., Wipfler, B., Wong, T. F., Wu, Q., Wu, G., Xie, Y., Yang, S., Yang, Q., Yeates, D. K. Yoshizawa, K., Zhang, Q., Zhang, R., Zhang, W., Zhang, Y., Zhao, J., Zhou, C., Zhou, L., Ziesmann, T., Zou, S., Li, Y., Xu, X., Zhang, Y., Yang, H., Wang, J. Wang, J., Kjer, K. M., and Zhou, X. (2014). Phylogenomics resolves the timing and pattern of insect evolution. *Science*, 346:763–767.

Sharma, P. P., Kaluziak, S. T., Pérez-Porro, A. R., González, V. L., Hormiga, G., Wheeler, W. C., and Giribet, G. (2014). Phylogenomic interrogation of Arachnida reveals systemic conflicts in phylogenetic signal. *Molecular Biology and Evolution*, 31:2963–2984.

Smith, S. A. and Dunn, C. W. (2008). Phyutility: a phyloinformatics tool for trees, alignments and molecular data. *Molecular Phylogenetics and Evolution*, 66:526–538.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1355v1 | CC-BY 4.0 Open Access | rec: 8 Sep 2015, publ: 8 Sep 2015