SOFTWARE

BuddySuite: Command-line toolkits for manipulating sequences, alignments, and phylogenetic trees

Stephen R Bond, Karl E Keat and Andreas D Baxevanis[*]

Abstract

Background: BuddySuite is a collection of four independent yet interrelated command-line programs that facilitate each step in the workflow of sequence discovery, curation, alignment, and phylogenetic reconstruction. Common sequence, alignment, and tree file formats are automatically detected and parsed, and nearly 100 routine tasks have been combined into this comprehensive suite of toolkits.

Results: The project has been implemented in Python 3 for use on UNIX-based systems. Installation is performed using a dedicated graphical installer or by cloning the development Git repository. All source code is freely available.

Conclusions: http://research.nhgri.nih.gov/software/BuddySuite

Supplementary: Documentation for each BuddySuite tool is available at http://tiny.cc/buddysuite_wiki

Keywords: sample; article; author

Background

Manipulation of biological sequence data is now a routine task within the life sciences, not just by bioinformaticians, but also by 'bench biologists' who are becoming increasingly savvy in applying computational methods to their own work. While there are excellent graphical platforms for organizing, visualizing, and manipulating these forms of data, it is often advantageous to interact with text files directly from the command line, especially when the size of datasets become even moderately large. Most common tasks can be accomplished with existing open source software, but it is usually necessary to bring together many different standalone tools to build a particular workflow. Such tools may be dependent on pre-defined file format specifications, have non-trivial installation requirements, and/or be difficult to extend or modify. While each of these issues is surmountable, particularly if one can write custom programs in any of the popular scripting languages (e.g., Perl, R, or Python), they do impose an entry barrier to those without a basic background in computer science. Furthermore, finding available tools can be non-trivial, as specialized programs are not generally well advertised or highly ranked by search engines. To address these issues we have developed BuddySuite, a unified set of command-line data manipulation tools that are easy to install, intuitively organized, and implemented in the popular programming language Python. The target audience for this software is those with a basic working knowledge of the standard POSIX shell (e.g., command-line terminals in Linux or Mac OS X) who routinely interact with sequence, alignment, or phylogenetic tree files.

Implementation

ADD: Table of tools

ADD: Installation via BioConda and PyPI

ADD: Full list of wrapped software

Command line user interface

BuddySuite is implemented in pure Python and includes four core command line programs: SeqBuddy, AlignBuddy, PhyloBuddy, and DatabaseBuddy. The first three accept sequence, alignment, or phylogenetic tree data as input, respectively, and flags are used to specify which tool to run. DatabaseBuddy, on the other hand, is intended to run primarily as a

[*]Correspondence: andy@mail.nih.gov

Computational and Statistical Genomics Branch, Division of Intramural Research, National Human Genome Research Institute, National Institutes of Health, 50 South Drive, 20892 Bethesda, USA

Full list of author information is available at the end of the article

'live shell', allowing the user to interactively search for and download sequence data stored in public databases (e.g., NCBI, UniProt, and Ensembl). The BuddySuite programs collectively contain 103 individual tools at the time of this writing, each with extended help and usage examples on the BuddySuite wiki (http://tiny.cc/buddysuite_wiki).

BuddySuite commands can be 'daisy-chained' together with the pipe character (|) to create more complex workflows as a single line in the terminal.

Application programing interface (API)

that can be accessed directly from the command line or as an application programming interface (API) For those interested in integrating BuddySuite functions into their own Python 3 scripts, the process is simplified by base classes in each module that handle many forms of input (including plain text, file paths or handles, and BioPython objects), then automate format detection and pre-processing. An object invoked from one of these base classes is the first parameter of all BuddySuite functions and is also the output in most cases.

File format parsing

File format detection is automated, and most of the formats with BioPython parsers are supported [1]. Another key feature of BuddySuite is robust sequence annotation management. Flat file formats such as GenBank and EMBL allow for rich annotation of sequence features, and these will be retained and/or adjusted by SeqBuddy and AlignBuddy tools as necessary. As an example, the AlignBuddy 'generate_alignment' tool can be used to invoke popular third party alignment programs such as MAFFT [2] on an annotated GenBank file; after completion, the new alignment will be returned in GenBank format with all original features re-mapped to account for newly introduced gaps.

Installation

Users of BuddySuite have several options for installing and updating the software. Stable release versions are available from the Python Package Index [3] (http://tiny.cc/buddysuite_pypi) and BioConda [4] (http://tiny.cc/buddysuite_bioconda), allowing for automated installation with the programs 'pip' or 'conda', respectively. The project is also hosted on a public GitHub [5] repository (http://tiny.cc/buddysuite_github) with an active development branch for continuous integration, allowing immediate access to all new features as they are built. Unit tests have been written to cover >95% of the codebase and continuous integration is monitored with Travis CI [6] (http://tiny.cc/buddysuite_travisci).

Dependencies

Python standard library packages have been used where possible to minimize licensing and version incompatibility issues, although the suite does depend heavily on BioPython [1]. Furthermore, PhyloBuddy uses DendroPy [7]. for much of its tree manipulation functionality and the ETE toolkit [8] to graphically display trees. A number of optional programs are also used by individual functions within the BuddySuite, such as BLAST [9], MAFFT [2], and RAxML [10]. These programs are not distributed with BuddySuite, so the user is responsible for their installation if they wish to use the functions that rely on them.

Error/usage reporting and contribution

Looking forward, the modular nature of BuddySuite makes it particularly well suited for continued growth. New tools are easily added to each existing module and new modules may be added to the suite. Instead of relying exclusively on active community input to identify bugs and drive future development, we have implemented an optional passive data collection program to monitor usage and crash reporting. Personally identifiable information is stripped before any data is transmitted to our FTP server, and a randomly generated identifier is assigned to new systems when BuddySuite is installed to estimate attrition rates.

Results and Discussion

Use-case examples

This is how you would do stuff with SeqBuddy etc. For example, after downloading the cDNA sequence for all members of a gene family with DatabaseBuddy, the records could be renamed, annotated, and translated to amino acids with SeqBuddy, converted to a multiple sequence alignment and trimmed of poorly aligned regions with AlignBuddy, and then PhyloBuddy could be used to estimate a phylogenetic tree, split any polytomies, and root on a particular set of taxa. Furthermore, third party programs that use any of the supported file formats can be seamlessly included in these pipelines.

Performance

SOFIA'S SECTION

Here are some graphs and tables showing how long each tool takes to run on different sized files (Write an automated method to get all the stats).

Similar bioinformatics toolkits

Describe the state of EMBOSS and BioPieces To keep the learning curve as shallow as possible, care has been taken to minimize the dependence of each tool on additional parameters and to infer user intent where possible. For example, the SeqBuddy 'find_restriction_sites' function is one of the most flexible in the Suite. It can accept three different types of arguments that control which enzymes are included in the search and how the output is formatted; all of these arguments are optional and they can be passed to the tool in any order. This flexibility is in contrast to the paradigm implemented in EMBOSS and BioPieces, which require extra flags to explicitly set all parameters. In cases such as this, where the argument type (e.g., integer or string) unambiguously identifies how it should be used by the tool, it is counter-productive to require that the user remember additional flags to explicitly mark each input.

Conclusions

BuddySuite has been designed from the ground up to be an intuitive, extensible, and unified platform for routine command-line tasks performed on sequence, alignment, and phylogenetic tree files. By implementing this project in the popular language Python and distributing it through GitHub, along with extensive documentation, we hope to gain community support to continue building BuddySuite into an even more comprehensive open-source solution.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Text for this section . . .

Acknowledgements

Text for this section . . .

References

1. Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., de Hoon, M.J.L.: Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics **25**(11), 1422–1423 (2009)
2. Katoh, K., Standley, D.M.: MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Molecular biology and evolution **30**(4), 772–780 (2013)
3. Python Package Index. https://pypi.python.org/
4. Bioconda. https://bioconda.github.io/
5. GitHub. https://github.com/
6. Travis CI. https://travis-ci.org/
7. Sukumaran, J., Holder, M.T.: DendroPy: a Python library for phylogenetic computing. Bioinformatics **26**(12), 1569–1571 (2010)
8. Huerta-Cepas, J., Dopazo, J., Gabaldón, T.: ETE: a python Environment for Tree Exploration. BMC bioinformatics **11**, 24 (2010)
9. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., Madden, T.L.: BLAST+: architecture and applications. BMC bioinformatics **10**, 421 (2009)
10. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics **22**(21), 2688–2690 (2006)
11. Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J.D., Higgins, D.G.: Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Molecular systems biology **7**(1), 539–539 (2011)
12. Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., McGettigan, P.A., McWilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J., Higgins, D.G.: Clustal W and Clustal X version 2.0. Bioinformatics **23**(21), 2947–2948 (2007)
13. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research **32**(5), 1792–1797 (2004)
14. Löytynoja, A., Vilella, A.J., Goldman, N.: Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. Bioinformatics **28**(13), 1684–1691 (2012)
15. Löytynoja, A., Goldman, N.: An algorithm for progressive multiple alignment of sequences with insertions. Proceedings of the National Academy of Sciences of the United States of America **102**(30), 10557–10562 (2005)
16. Price, M.N., Dehal, P.S., Arkin, A.P.: FastTree 2–approximately maximum-likelihood trees for large alignments. PloS one **5**(3), 9490 (2010)
17. Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., Gascuel, O.: New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Systematic biology **59**(3), 307–321 (2010)

Figures

Figure 1 Sample figure title. A short description of the figure content should go here.

Figure 2 Sample figure title. Figure legend text.

Tables

Table 1 File format reading (R) and writing (W) support provided by each BuddySuite program.

Format	SeqBuddy	AlignBuddy	PhyloBuddy
Clustal	R & W†	R & W	None
EMBL‡	R & W	R†/ W	None
FASTA	R & W	R†/ W	None
GenBank‡	R & W	R†/ W	None
Nexus	R & W†	R & W	R & W
Newick	None	None	R & W
NeXML	None	None	R & W
PHYLIP (interleaved)	R & W†	R & W	None
PHYLIP (sequential)	R & W†	R & W	None
SeqXML	R & W	None	None
Stockholm	R & W†	R & W	None
Swissprot‡	R only	None	None

†All sequences must be the same length
‡Supports rich sequence annotation

Additional Files

Additional file 1 — Sample additional file title
Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title
Additional file descriptions text.

Table 2 List of optional third party software that BuddySuite programs can interact with.

BuddySuite program	3rd-party program	Reference
SeqBuddy	BLAST	[9]
AlignBuddy	Clustal Omega	[11]
	ClustalW2	[12]
	MAFFT	[2]
	MUSCLE	[13]
	PAGAN	[14]
	PRANK	[15]
PhyloBuddy	FastTree	[16]
	RAxML	[10]
	PhyML	[17]