SOFTWARE

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

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

Abstract

Background: The ability to manipulate sequence, alignment, and phylogenetic tree files has become an increasingly important skill in the life sciences, whether it be to generate summary information about those files or to prepare them for further downstream analysis. The command-line is generally the most powerful environment for interacting with these resources, especially as the files become even moderately large, but there has been little focus on developing or maintaining general purpose toolkits in recent years.

Results: 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 over 100 routine tasks have been implemented in this comprehensive suite of toolkits. The project has been engineered to easily accommodate the addition of new tools, written in the popular programming language Python, and hosted on the Python Package Index and GitHub to maximize accessibility. Documentation for each BuddySuite tool, including usage examples, is available at http://tiny.cc/buddysuite_wiki.

Conclusions: All software is open source and freely available without restriction. http://research.nhgri.nih.gov/software/BuddySuite

Keywords: software; command line; sequence; alignment; phylogenetic tree; python; toolkits

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., bash, 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 UNIX shell environment who routinely interact with sequence, alignment, or phylogenetic tree files.

*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

Implementation

BuddySuite is a set of Python 3 libraries and command line applications developed for use on all major operating systems (Windows 7+, Mac OSX, and

Linux), which leverages the sequence and phylogenetic tree processing capabilities of Biopython [1], Environment of Tree Exploration 3 (ETE3) [2], and Dendropy [3]. The project is free and open-source, versioned on Github [4] and the Python Package Index [5] (PyPI), and unit tested at a code coverage of over 95%.

Installation

Stable release vesions of BuddySuite can be installed directly from PyPI using the popular package manager 'pip', and development versions from GitHub are also easily installed using the provided setup script. While optional, we suggest that users also run the BuddySuite configuration script after installation (from the command line, "$: buddysuite -setup"). Doing so will create directories for caching data on the users' system and will register an email address for the tools that interact with public databases (to prevent possible IP blocking). Dependencies are limited to packages available through PyPI, although there are a number of optional third-party programs that can be accessed through BuddySuite; these include BLAST [6] for comparing sequences, multiple sequence alignment packages like MAFFT [7], and phylogenetic inference packages like RAxML [8]. These programs are not necessary for the general operation of the BuddySuite modules, however, so installation is the users' responsibility. The list of third-party tools that BuddySuite wraps is itemized in Table 1.

Command line user interface

The four core command line programs distributed with BuddySuite are SeqBuddy, AlignBuddy, PhyloBuddy, and DatabaseBuddy. The first three accept sequence, alignment, or phylogenetic tree data as input, respectively, using flags to switch among the tools available in each program. All output is printed directly to the terminal window by default and each module adheres to the UNIX convention of accepting piped data, allowing individual tools to be 'daisy-chained' into more complex workflows. DatabaseBuddy, on the other hand, is intended to run primarily as a 'live shell', allowing the user to interactively search and download sequence data stored in the NCBI, UniProt, and Ensembl public databases. The BuddySuite programs collectively contain 104 individual tools at the time of this writing. The '-h' and '–help' flags will list all available tools in a given module, along with basic usage instructions, while extended help and fully worked examples are available on the BuddySuite wiki (http://tiny.cc/buddysuite_wiki).

Application programing interface (API)

Each BuddySuite module has a core 'Buddy' class that automatically handles a variety of input types (e.g., plain test, file paths or handles, or a list of Biopython objects), performs all necessary file format processing, and exposes methods for managing and writing the resultant sequence or tree records. All of the API functions in each library accept these 'Buddy' objects as input and generally return them as output, thus providing a standardizing interface that facilitates interoperability among functions and makes the addition new functions very easy. Once installed, the BuddySuite libraries can be imported into third-party Python scripts normally.

Error reporting and usage statistics

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 system to monitor usage and to report crashes. This data is transmitted to an FTP server after all personally identifiable information has been stripped away. This also allows us to inform users of available bug fixes, as a crash traceback can be combined with a module's version number to create a unique identification hash; once identified, these hashes are stored in the Git repository along with their status (i.e., pending or resolved). If an update is available that will resolve a particular issue, the user will be informed at the time of the crash.

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).

The unique features of BuddySuite

The European Molecular Biology Open Software Suite (EMBOSS) and Biopieces are the most comprehensive general-purpose opensource bioinformatics toolkits currently available for the command line, and while both are excellent software, the development of BuddySuite is justified by a number of key differences. In particular is the switch away from the 'one program per function' paradigm that EMBOSS and Biopeices employ (each suite contains about 200 separate programs). BuddySuite groups all functions related to a particular data type together into specific modules and uses flags to differentiate among them; this reduces the potential for naming collisions on a user's system PATH. BuddySuite is also the only general-purpose toolkit implemented entirely in pure Python. This is unlike EMBOSS, which must be compiled primarily from C, and Biopieces, which relies on Python, Perl, and Ruby. While there is a performance cost when running an interpreted language like Python, it makes installation easier and it reduces the entry barrier for public contribution to the project. Python and R have now emerged as the main prototyping and scripting languages in the life sciences [REF - Comparing languages], largely due to the growing number of researchers who are learning to program for the general purpose of data wrangling [REF - SWC]. This positions BuddySuite as a more approachable option for users who wish to implement custom functionality.

To keep the learning curve as shallow as possible, care has been taken to minimize the number of parameters each tool depends upon and to use duck typing to infer user intent. For example, the SeqBuddy 'find_restriction_sites' function is one of the most flexible in the Suite; it can accept three different argument types that control what enzymes are included in the search and how the output is formatted, yet all of these arguments are optional and can be passed to the tool in any order. This flexibility is in contrast to EMBOSS and BioPieces, which generally require extra flags to explicitly set all parameters. When the argument type (e.g., integer or string) unambiguously identifies how it should be used by the tool, we believe it is counter-productive to impose positional constraints or additional flags. Furthermore, file format detection is fully automated. Any number of sequence, alignment, or phylogenetic tree files can be passed into their respective BuddySuite program, in any combination of supported formats, and the records will be parsed seamlessly (see table 2 for a list of supported formats). Bioinformatics software is rarely this flexible in terms of user input. For specialized tools that are intended to be called from scripts in larger pipelines, explicitly setting parameters can increase methodological clarity and is less of a burden because they are not typed repeatedly. For a general purpose tool like BuddySuite, however, where the user is intended to interact with their data dynamically on the command line, minimizing key-strokes is crucial.

Perhaps the greatest advantage BuddySuite has over other tools is its handling of annotations. Rich flat-file formats like GenBank and EMBL support sequence feature annotation, but the information is generally discarded by the EMBOSS programs and the Biopieces suite is unable to write these formats. SeqBuddy and AlignBuddy are both aware of features in the sequence records they process, and will update those annotations when sequences are modified. For example, if a group of DNA sequences are translated to protein with SeqBuddy, the relative positions of each feature will be scaled by one third to account for the conversion of codons to amino acids. If those proteins are then passed to AlignBuddy to create a multiple sequence alignment, the features will be adjusted again to account for any gaps that are introduced (see figure X). Unfortunately, support for the Generic Feature Format (GFF3) specification [REF] is not currently implemented in BioPython, so is also not available in BuddySuite at the time of writing. While GFF3 support is planned for a future release, users are currently encouraged to write annotated sequences to GenBank or EMBL (this includes alignments, as BuddySuite will respect gap characters in these formats).

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.

Every effort has been made to ensure that the user experience is easy and enjoyable, By implementing this project in the popular language Python and distributing it through PyPI and 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

SRB is the lead developer of BuddySuite and wrote the manuscript, KEK contributed significantly to the code base, SNB generated the performance statistics for FIG.X, ADB was involved in the design and coordination of the project. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank the community members who contributed code to this project, big or small. It takes a village.

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. Huerta-Cepas, J., Serra, F., Bork, P.: ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. Molecular biology and evolution **33**(6), 1635–1638 (2016)
3. Sukumaran, J., Holder, M.T.: DendroPy: a Python library for phylogenetic computing. Bioinformatics **26**(12), 1569–1571 (2010)
4. GitHub. https://github.com/
5. Python Package Index. https://pypi.python.org/
6. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., Madden, T.L.: BLAST+: architecture and applications. BMC bioinformatics **10**, 421 (2009)
7. 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)
8. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics **22**(21), 2688–2690 (2006)
9. 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)
10. 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)
11. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research **32**(5), 1792–1797 (2004)
12. 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)
13. 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)
14. Price, M.N., Dehal, P.S., Arkin, A.P.: FastTree 2–approximately maximum-likelihood trees for large alignments. PloS one **5**(3), 9490 (2010)
15. 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

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

BuddySuite program	3rd party program	Reference
SeqBuddy	BLAST	[6]
AlignBuddy	Clustal Omega	[9]
	ClustalW2	[10]
	MAFFT	[7]
	MUSCLE	[11]
	PAGAN	[12]
	PRANK	[13]
PhyloBuddy	FastTree	[14]
	RAxML	[8]
	PhyML	[15]

Table 2 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