

## RESEARCH

# 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](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 stand-alone 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 back-ground 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 at least 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

BuddySuite is written exclusively in Python 3 and is composed of four separate 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 ‘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 modules collectively contain 95 individual tools at the time of this writing, each with ex-

\*Correspondence: [andy@mail.nih.gov](mailto: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

tended help and usage examples on the BuddySuite wiki (<http://tiny.cc/buddysuite-wiki>). File format detection is automated, and most of the formats with BioPython parsers are supported (Cock *et al.*, 2009). 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 when arguments are provided.

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

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 (Katoh and Standley, 2013) 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.

To facilitate easy installation, the entire suite is released as an executable (BuddySuite.py) that bundles all dependencies with a graphical installation script. The user is guided through configuration options and allowed to select which of the BuddySuite modules to add to their system. On non-graphical systems the installation script will run in a command-line mode. The same installer can be used later to modify optional parameters or completely uninstall the software. The source code is also available for download on GitHub (<http://tiny.cc/buddysuite>), although dependencies will need to be installed manually if the programs are accessed this way.

Python standard library packages have been used where possible to minimize licensing and version incompatibility issues, although the suite does depend heavily on BioPython (Cock *et al.*, 2009). Furthermore, PhyloBuddy uses DendroPy (Sukumaran and Holder, 2010) for much of its tree manipulation functionality and version 3.0 of the ETE toolkit (Huerta-

Cepas *et al.*, 2010) to graphically display trees. A number of optional programs are also used by individual functions within the BuddySuite, such as BLAST (Camacho *et al.*, 2009), MAFFT (Katoh and Standley, 2013), and RAxML (Stamatakis, 2006). Most of 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.

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.

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.

In conclusion, 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

#### Figures

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

**Figure 2** Sample figure title. Figure legend text.

**Table 1** Sample table title. This is where the description of the table should go.

	B1	B2	B3
A1	0.1	0.2	0.3
A2	...	..	.
A3	..	.	.

**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.