Biopython: freely available Python tools for computational molecular biology and bioinformatics

Peter J. A. Cock [1]*, Tiago Antao [2], Jeffrey T. Chang [3], Brad A. Chapman [4], Cymon J. Cox [5], Andrew Dalke [6], Iddo Friedberg [7], Thomas Hamelryck [8], Frank Kauff [9], Bartek Wilczynski [10], and Michiel J. L. de Hoon [11]

[1]Plant Pathology, SCRI, Invergowrie, Dundee DD2 5DA, UK, [2]Liverpool School of Tropical Medicine, Liverpool L3 5QA, UK, [3]Institute for Genome Sciences and Policy, Duke University Medical Center, Durham, NC, USA, [4]Department of Molecular Biology, Simches Research Center, Boston, MA 02114, USA, [5]Centro de Ciências do Mar, Universidade do Algarve, Faro, Portugal, [6]Andrew Dalke Scientific, AB, Gothenburg, Sweden, [7]California Institute for Telecommunications and Information Technology, University of California, San Diego, 9500 Gilman Dr., La Jolla, CA 92093-0446, USA, [8]Bioinformatics Center, Department of Biology, University of Copenhagen, Ole Maaloes Vej 5, 2200 Copenhagen N, Denmark, [9]Molecular Phylogenetics, Department of Biology, TU Kaiserslautern, 67653 Kaiserslautern, Germany, [10]EMBL Heidelberg, Meyerhofstraße 1, 69117 Heidelberg, Germany, & Institute of Informatics, University of Warsaw, Poland, and [11]RIKEN Omics Science Center, 1-7-22 Suehiro-cho, Tsurumi-ku, Yokohama-shi, Kanagawa-ken, 230-0045, Japan

Associate Editor Prof. Dmitrij Frishman

ABSTRACT

Summary: The Biopython project is a mature open source international collaboration of volunteer developers, providing Python libraries for a wide range of bioinformatics problems. Biopython includes modules for reading and writing different sequence file formats and multiple sequence alignments, dealing with 3D macromolecular structures, interacting with common tools such as BLAST, ClustalW, and EMBOSS, accessing key online databases, as well as providing numerical methods for statistical learning.

Availability: Biopython is freely available, with documentation and source code, at www.biopython.org under the Biopython license.

Contact: All queries should be directed to the Biopython mailing lists, see www.biopython.org/wiki/Mailing_lists.

1 INTRODUCTION

Python (www.python.org) and Biopython are freely available open source tools, available for all the major operating systems. Python is a very high-level programming language, in widespread commercial and academic use. It features an easy to learn syntax, object-oriented programming capabilities, and a wide array of libraries. Python can interface to optimised code written in C, C++, or even FORTRAN, and together with the Numerical Python project `numpy` (Oliphant, 2006), makes a good choice for scientific programming (Oliphant, 2007). Python has even been used in the numerically demanding field of molecular dynamics (Hinsen, 2000). There are also high quality plotting libraries such as `matplotlib` (matplotlib.sourceforge.net) available.

Since its founding in 1999 (Chapman and Chang, 2000), Biopython has grown into a large collection of modules, described briefly below, intended for computational biology or bioinformatics programmers to use in scripts or incorporate into their own software. Our website lists over 100 publications using or citing Biopython.

The Open Bioinformatics Foundation (OBF, www.open-bio.org) hosts our website, source code repository, bug tracking database, and email mailing lists, and also supports the related BioPerl (Stajich *et al.*, 2002), BioJava (Holland *et al.*, 2008), BioRuby (www.bioruby.org), and BioSQL (www.biosql.org) projects.

2 BIOPYTHON FEATURES

The `Seq` object is Biopython's core sequence representation. It behaves very much like a Python string but with the addition of an alphabet (allowing explicit declaration of a protein sequence for example) and some key biologically relevant methods. For example,

```
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import generic_dna
>>> gene = Seq("ATGAAAGCAATTTTCGTACTG"
...            "AAAGGTTGGTGGCGCACTTGA",
...            generic_dna)
>>> print gene.transcribe()
AUGAAAGCAAUUUUCGUACUGAAAGGUUGGUGGCGCACUUGA
>>> print gene.translate(table=11)
MKAIFVLKGWWRT*
```

Sequence annotation is represented using `SeqRecord` objects which augment a `Seq` object with properties such as the record name, identifier and description, and space for additional key/value

*to whom correspondence should be addressed

Table 1. Selected `Bio.SeqIO` or `Bio.AlignIO` file formats.

Format	R/W	Name and reference
`fasta`	R+W	FASTA (Pearson and Lipman, 1988)
`genbank`	R+W	GenBank (Benson *et al.*, 2007)
`embl`	R	EMBL (Kulikova *et al.*, 2006)
`swiss`	R	Swiss-Prot/TrEMBL or UniProtKB (The UniProt Consortium, 2007)
`clustal`	R+W	Clustal W (Thompson *et al.*, 1994)
`phylip`	R+W	PHYLIP (Felsenstein, 1989)
`stockholm`	R+W	Stockholm or Pfam (Bateman *et al.*, 2004)
`nexus`	R+W	NEXUS (Maddison *et al.*, 1997)

Where possible, our format names (column one) match BioPerl and EMBOSS (Rice *et al.*, 2000). Column two denotes support for reading (R) and writing (W).

terms. The `SeqRecord` can also hold a list of `SeqFeature` objects which describe sub-features of the sequence with their location and their own annotation.

The `Bio.SeqIO` module provides a simple interface for reading and writing biological sequence files in various formats (Table 1), where regardless of the file format, the information is held as `SeqRecord` objects. `Bio.SeqIO` interprets multiple sequence alignment file formats as collections of equal length (gapped) sequences. Alternatively, `Bio.AlignIO` works directly with alignments, including files holding more than one alignment (e.g., re-sampled alignments for bootstrapping, or multiple pairwise alignments). Related module `Bio.Nexus`, developed for Kauff *et al.* (2007), supports phylogenetic tools using the NEXUS interface (Maddison *et al.*, 1997), or the Newick standard tree format.

Modules for a number of online databases are included, such as the NCBI Entrez Utilities, ExPASy, InterPro, KEGG, and SCOP. `Bio.Blast` can call the NCBI's online Blast server or a local standalone installation, and includes a parser for their XML output. Biopython has wrapper code for other command line tools too, such as ClustalW and EMBOSS. The `Bio.PDB` module provides a PDB file parser, and functionality related to macromolecular structure (Hamelryck and Manderick, 2003). Module `Bio.Motif` provides support for sequence motif analysis (searching, comparing and de novo learning). Biopython's graphical output capabilities were recently significantly extended by the inclusion of GenomeDiagram (Pritchard *et al.*, 2006).

Biopython contains modules for supervised statistical learning, such as Bayesian methods and Markov models, as well as unsupervised learning, such as clustering (De Hoon *et al.*, 2004).

The population genetics module provides wrappers for GENEPOP (Rousset, 2007), coalescent simulation via SIMCOAL2 (Laval and Excoffier, 2004), and selection detection based on a well-evaluated F_{st}-outlier detection method (Beaumont and Nichols, 1996).

BioSQL (www.biosql.org) is another OBF supported initiative, a joint collaboration between BioPerl, Biopython, BioJava, and BioRuby to support loading and retrieving annotated sequences to and from an SQL database using a standard schema. Each project provides an object-relational mapping (ORM) between the shared schema and its own object model (a `SeqRecord` in Biopython). As an example, *x*BASE (Chaudhuri and Pallen, 2006) uses BioSQL with both BioPerl and Biopython.

3 CONCLUSIONS

Biopython is a large open-source API used in both bioinformatics software development and in everyday scripts for common bioinformatics tasks. The homepage www.biopython.org provides access to the source code, documentation, and mailing lists. The features described herein are only a subset, potential users should refer to the tutorial and API documentation for further information.

ACKNOWLEDGEMENTS

The OBF hosts and supports the project. Fundacao para a Ciencia e Tecnologia (Portugal) grant SFRH/BD/30834/2006 supported TA. The many Biopython contributors over the years are warmly thanked, a list too long to be reproduced here.

REFERENCES

Ascher, D. *et al.* (2001) An Open Source Project : Numerical Python, Lawrence Livermore National Laboratory.

Chapman, B. and Chang, J. (2000) Biopython: Python tools for computational biology. *ACM SIGBIO Newslett.*, **20**, 15-19.

Chaudhuri, R.R. and Pallen, M.J. (2006) xBASE, a collection of online databases for bacterial comparative genomics. *Nucleic Acids Res.*, Database Issue, **34**, D335D337.

Bateman, A. *et al.* (2004) The Pfam Protein Families Database. *Nucleic Acids Res.*, Database Issue **32**, D138-D141.

Beaumont, M.A. and Nichols, R.A. (1996) Evaluating loci for use in the genetic analysis of population structure. *Proc. R. Soc. Lond. B*, **263**, 1619-1626.

Benson, D.A. *et al.* (2007) GenBank. *Nucleic Acids Res.*, Database Issue **35**, D21-5.

Felsenstein, J. (1989) PHYLIP - Phylogeny Inference Package (Version 3.2), *Cladistics*, **5**, 164-166.

Hamelryck, H. and Manderick, B. (2003) PDB file parser and structure class implemented in Python. *Bioinformatics*, **19**(17), 2308-2310.

Hinsen, K. (2000) The Molecular Modeling Toolkit: A New Approach to Molecular Simulations. *J. Comp. Chem.*, **21**, 79-85.

Holland *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**(18), 2096-2097.

De Hoon, M.J.L. *et al.* (2004) Open source clustering software. *Bioinformatics*, **20**(9), 1453-1454.

Kauff, F. *et al.* (2007) WASABI: An automated sequence processing system for multi-gene phylogenies. *Syst. Biol.* **56**(3): 523-531.

Kulikova, T. *et al.* (2006) EMBL Nucleotide Sequence Database in 2006, *Nucleic Acids Res.*, Database Issue **35**, D16-D20.

Lavel, G. and Excoffier, L. (2004) SIMCOAL 2.0: a program to simulate genomic diversity over large recombining regions in a subdivided population with a complex history. *Bioinformatics*, **20**(15), 2485-2487.

Maddison, D.R., Swofford, D.L., and Maddison, W.P. (1997) NEXUS: An extensible file format for systematic information. *Syst. Biol.*, **46**(4), 590-621.

Oliphant, T.E. (2006) Guide to NumPy, Trelgol Publishing, USA.

Oliphant, T.E. (2007) Python for Scientific Computing, *Comput. Sci. Eng.*, **9**(3), 10-20.

Pearson, W.R. and Lipman, D.J. (1988) Improved Tools for Biological Sequence Analysis, *PNAS*, **85**, 2444-2448.

Pritchard, L. *et al.* (2006) GenomeDiagram: A Python Package for the Visualisation of Large-Scale Genomic Data, *Bioinformatics* **22**(5), 616-617.

Rice, P. *et al.* (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, **16**(6), 276-277.

Rousset, (2007) GENEPOP '007: a complete re-implementation of the GENEPOP software for Windows and Linux. *Molecular Ecology Resources*, **8**(1), 103-106.

Stajich, J.E. *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res*, **12**(10), 1611-8.

Thompson, J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673-4680.

The UniProt Consortium (2007) The Universal Protein Resource (UniProt), *Nucleic Acids Res.*, **35**, D193-D197.