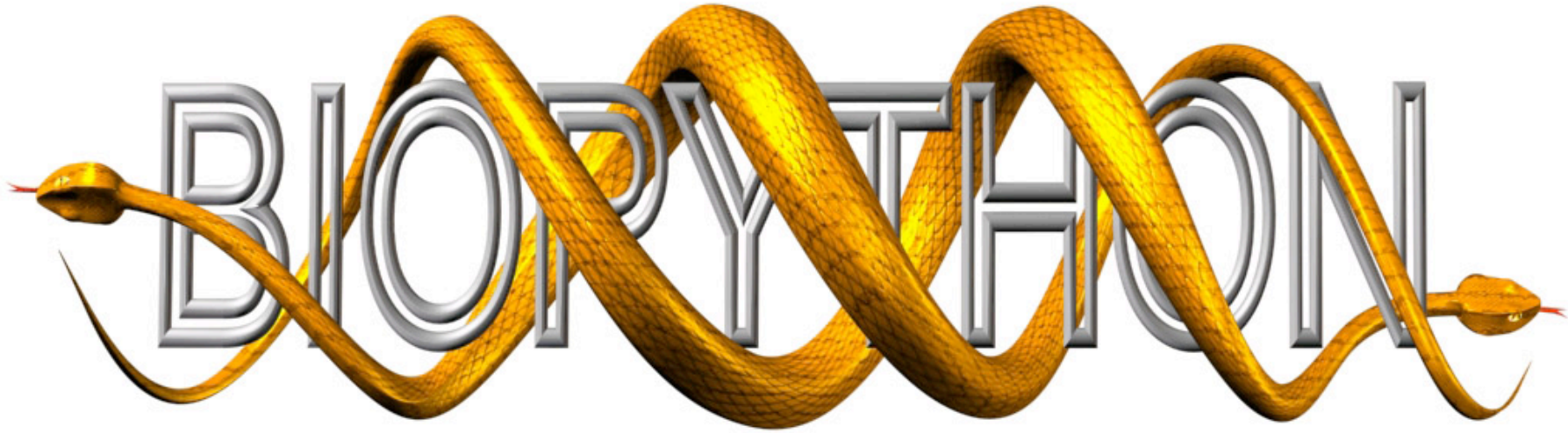


Biopython Project Update

Peter Cock, Plant Pathology, SCRI, Dundee, UK

10th Annual Bioinformatics Open Source Conference (BOSC)

Stockholm, Sweden, 28 June 2009

scri
living technology

Contents

- Brief introduction to Biopython & history
- Releases since BOSC 2008
- Current and future projects
- CVS, git and github
- BoF hackathon and tutorial at BOSC 2009

scri
living technology

Biopython

- Free, open source library for bioinformatics
- Supported by Open Bioinformatics Foundation
- Runs on Windows, Linux, Mac OS X, etc
- International team of volunteer developers
- Currently about three releases per year
- Extensive “Biopython Tutorial & Cookbook”
- See www.biopython.org for details

scri
living technology

Biopython's Ten Year History

1999	•	Started by Jeff Chang & Andrew Dalke
2000	•	First release, Biopython 0.90
2001	•	Biopython 1.00, "semi-complete"
…	•	Biopython 1.10, …, 1.41
2007	•	Biopython 1.43 (Bio.SeqIO), 1.44
2008	•	Biopython 1.45, 1.47, 1.48, 1.49
2009	•	Biopython 1.50, 1.51*beta*
	•	OA Publication, Cock *et al.*

scri
living technology

Biopython Publication – Cock *et al*. 2009

BIOINFORMATICS APPLICATIONS NOTE

Vol. 25 no. 11 2009, pages 1422–1423
doi:10.1093/bioinformatics/btp163

Sequence analysis

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[10,11] and Michiel J. L. de Hoon[12]

[1]Plant Pathology, SCRI, Invergowrie, Dundee, DD2 5DA, [2]Liverpool School of Tropical Medicine, Liverpool, L3 5QA, UK, [3]Institute for Genome Sciences and Policy, Duke University Medical Center, Durham, NC, [4]Department of Molecular Biology, Simches Research Center, Massachusetts General Hospital, 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, UK, [10]EMBL Heidelberg, Meyerhofstraße 1, 69117 Heidelberg, Germany, [11]Institute of Informatics, University of Warsaw, Poland and [12]RIKEN Omics Science Center, 1-7-22 Suehiro-cho, Tsurumi-ku, Yokohama-shi, Kanagawa-ken, 230-0045, Japan

Received and revised on March 11, 2009; accepted on March 16, 2009

Advance Access publication March 20, 2009

Associate Editor: 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; peter.cock@scri.ac.uk.

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 optimized 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 web site lists over 100 publications using or citing Biopython.

The Open Bioinformatics Foundation (OBF, www.open-bio.org) hosts our web site, 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 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.

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 'Format') match BioPerl and EMBOSS (Rice *et al*., 2000). Column 'R/W' denotes support for reading (R) and writing (W).

3 CONCLUSIONS

Biopython is a large open-source application programming interface (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. The many Biopython contributors over the years are warmly thanked, a list too long to be reproduced here.

Funding: Fundacao para a Ciencia e Tecnologia (Portugal) (grant SFRH/BD/30834/2006 to T.A.).

Conflict of Interest: none declared.

REFERENCES

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.*, **34**, D335–D337.

Bateman,A. *et al*. (2004) The Pfam protein families database. *Nucleic Acids Res.*, **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.*, **35**, D21–D25.

Felsenstein,J. (1989) PHYLIP – phylogeny inference package (Version 3.2). *Cladistics*, **5**, 164–166.

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

Hinsen,K. (2000) The molecular modeling toolkit: a new approach to molecular simulations. *J. Comp. Chem.*, **21**, 79–85.

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

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

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

Kulikova,T. *et al*. (2006) EMBL nucleotide sequence database in 2006. *Nucleic Acids Res.*, **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**, 2485–2487.

Maddison,D.R. *et al*. (1997) NEXUS: an extensible file format for systematic information. *Syst. Biol.*, **46**, 590–621.

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

Oliphant,T.E. (2007) Python for Scientific Computing. *Comput. Sci. Eng.*, **9**, 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**, 616–617.

Rice,P. *et al*. (2000) EMBOSS: the European molecular biology open software suite. *Trends Genet.*, **16**, 276–277.

Rousset,F. (2007) GENEPOP '007: a complete re-implementation of the GENEPOP software for Windows and Linux. *Mol. Ecol. Res.*, **8**, 103–106.

Stajich,J.E. *et al*. (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.

The UniProt Consortium. (2007) The universal protein resource (UniProt). *Nucleic Acids Res.*, **35**, D193–D197.

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.

*To whom correspondence should be addressed.

N.B. Open Access!

scri living technology

November 2008 – Biopython 1.49

- Support for Python 2.6
- Switched from “Numeric” to “NumPy” (important Numerical library for Python)
- More biological methods on core Seq object

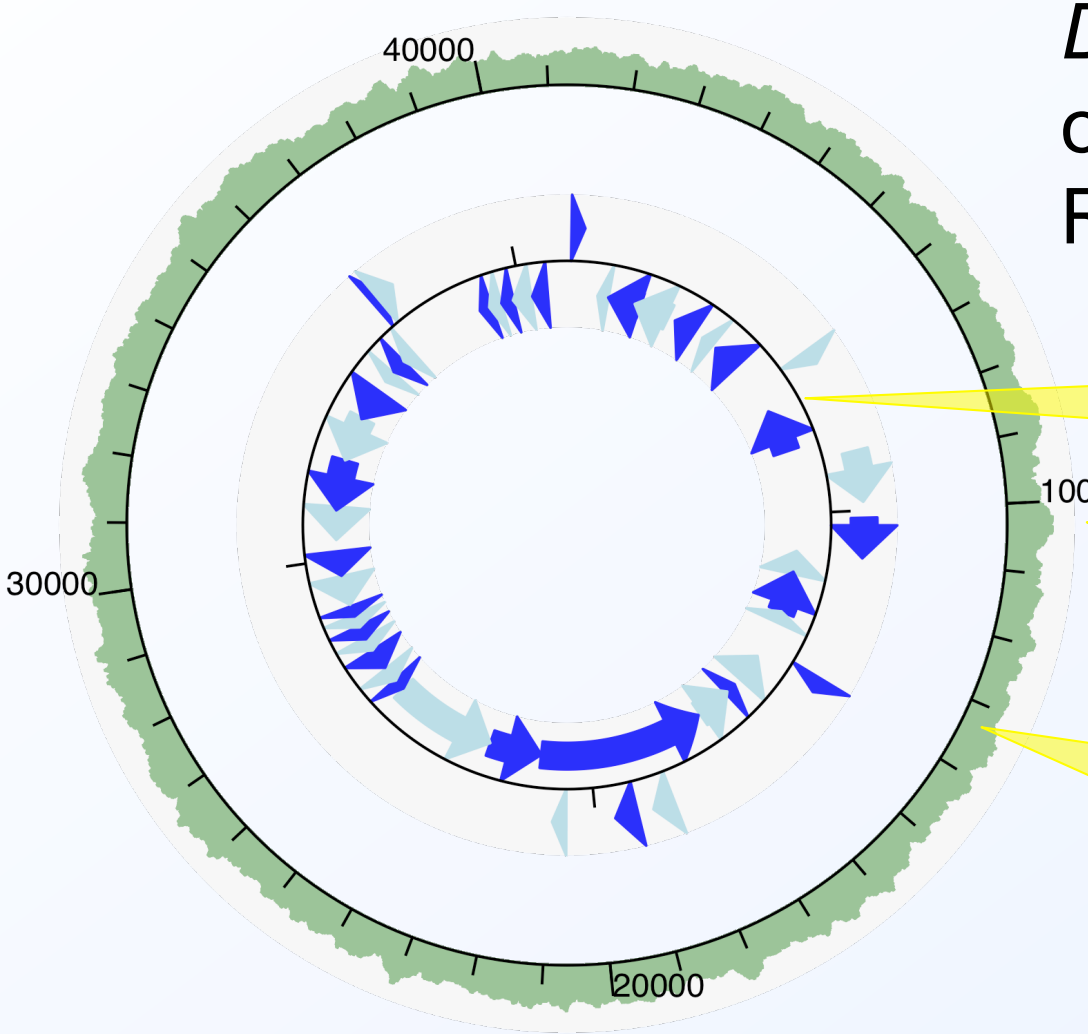
scri
living technology

April 2009 – Biopython 1.50

- New Bio.Motif module for sequence motifs (to replace Bio.AliceAce and Bio.MEME)
- Support for QUAL and FASTQ in Bio.SeqIO (important NextGen sequencing formats)
- Integration of GenomeDiagram for figures (Pritchard *et al*. 2006)

scri
living technology

Biopython 1.50 includes GenomeDiagram



De novo assembly of 42kb phage from Roche 454 data

"Feature Track" showing ORFs

Scale tick marks

"Barchart Track" of read depth (~100, scale max 200)

scri
living technology

Reading a FASTA file with Bio.SeqIO

```
>FL3B07415JACDX
TTAATTTTATTTTGTCGGCTAAAGAGATTTTTAGCTAAACGTTCAATTGCTTTAGCTGAA
GTACGAGCAGATACTCCAATCGCAATTGTTTCTTCATTTAAAATTAGCTCGTCGCCACCT
TCAATTGGAAATTTATAATCACGATCTAACCAGATTGGTACATTATGTTTTGCAAATCTT
GGATGATATTTAATGATGTACTCCATGAATAATGATTCACGTCTACGCGCTGGTTCTCTC
ATCTTATTTATCGTTAAGCCA
>FL3B07415I7AFR
...
```

```
from Bio import SeqIO

for rec in SeqIO.parse(open("phage.fasta"), "fasta") :
    print rec.id, len(rec.seq), rec.seq[:10]+"..."
```

```
FL3B07415JACDX 261 TTAATTTTAT...
FL3B07415I7AFR 267 CATTAACTAA...
FL3B07415JCAY5 136 TTTCTTTTCT...
FL3B07415JB41R 208 CTCTTTTATG...
FL3B07415I6HKB 268 GGTATTTGAA...
FL3B07415I63UC 219 AACATGTGAG...
...
```

Focus on the filename and format (“fasta”)…

scri living technology

Reading a FASTQ file with Bio.SeqIO

```
@FL3BO7415JACDX
TTAATTTTATTTTGTCGGCTAAAGAGATTTTTAGCTAAACGTTCAATTGCTTTAGCTGAAGTACGAGCAGATACTCCAATCGCAATTGTTTCTTC
ATTTAAAATTAGCTCGTCGCCACCTTCAATTGGAAATTTATAATCACGATCTAACCAGATTGGTACATTATGTTTTGCAAATCTTGGATGATATT
TAATGATGTACTCCATGAATAATGATTCACGTCTACGCGCTGGTTCTCTCATCTTATTTATCGTTAAGCCA
+
BBBB2262=1111FFGGGHHHHIIIIIIIIIIIIIIIIIIIIIIFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGGFFFFFFFFFFFFFFFGB
BBCFFFFFFFFFFFFFFFFFFFFFGGGGGGGIIIIIIIGGGIIIGGGIIGGGG@AAAAA?===@@@???
@FL3BO7415I7AFR
...
```

```
from Bio import SeqIO

for rec in SeqIO.parse(open("phage.fastq"), "fastq") :
    print rec.id, len(rec.seq), rec.seq[:10]+"..."
    print rec.letter_annotations["phred_quality"][:10], "..."
```

```
FL3BO7415JACDX 261 TTAATTTTAT...
[33, 33, 33, 33, 17, 17, 21, 17, 28, 16] ...
FL3BO7415I7AFR 267 CATTAACTAA...
[37, 37, 37, 37, 37, 37, 37, 37, 38, 38] ...
FL3BO7415JCAY5 136 TTTCTTTTCT...
[37, 37, 36, 36, 29, 29, 29, 29, 36, 37] ...
FL3BO7415JB41R 208 CTCTTTTATG...
[37, 37, 37, 38, 38, 38, 38, 38, 37, 37] ...
FL3BO7415I6HKB 268 GGTATTTGAA...
[37, 37, 37, 37, 34, 34, 34, 37, 37, 37] ...
FL3BO7415I63UC 219 AACATGTGAG...
[37, 37, 37, 37, 37, 37, 37, 37, 37, 37] ...
...
```

Just filename and format changed ("fasta" to "fastq")

scri living technology

June 2009 – Biopython 1.51 *beta*

- Support for Illumina 1.3+ FASTQ files (in addition to Sanger FASTQ and older Solexa/Illumina FASTQ files)
- Faster parsing of UniProt/SwissProt files
- Bio.SeqIO now writes feature table in GenBank output

Already being used at SCRI for genome annotation, e.g. with RAST and Artemis

scri
living technology

Google Summer of Code Projects

- Eric Talevich - Parsing and writing phyloXML
  - Mentors Brad Chapman & Christian Zmasek
- Nick Matzke - Biogeographical Phylogenetics
  - Mentors Stephen Smith, Brad Chapman & David Kidd
- Hosted by NESCent Phyloinformatics Group
- Code development on github branches...

Other Notable Active Projects

- Brad Chapman – GFF parsing
- Tiago Antão – Population genetics statistics
- Peter Cock – Parsing Roche 454 SFF files (with Jose Blanca, co-author of sff_extract)
- Plus other ongoing refinements and documentation improvements

scri
living technology

Distributed Development

- Currently work from a stable branch in CVS
- CVS master is mirrored to github.com
- Several sub-projects are being developed on github branches (in public)
- This is letting us get familiar with git & github
- Suggested plan is to switch from CVS to git summer 2009 (still hosted by OBF), continue to push to github for public collaboration

scri
living technology

Acknowledgements

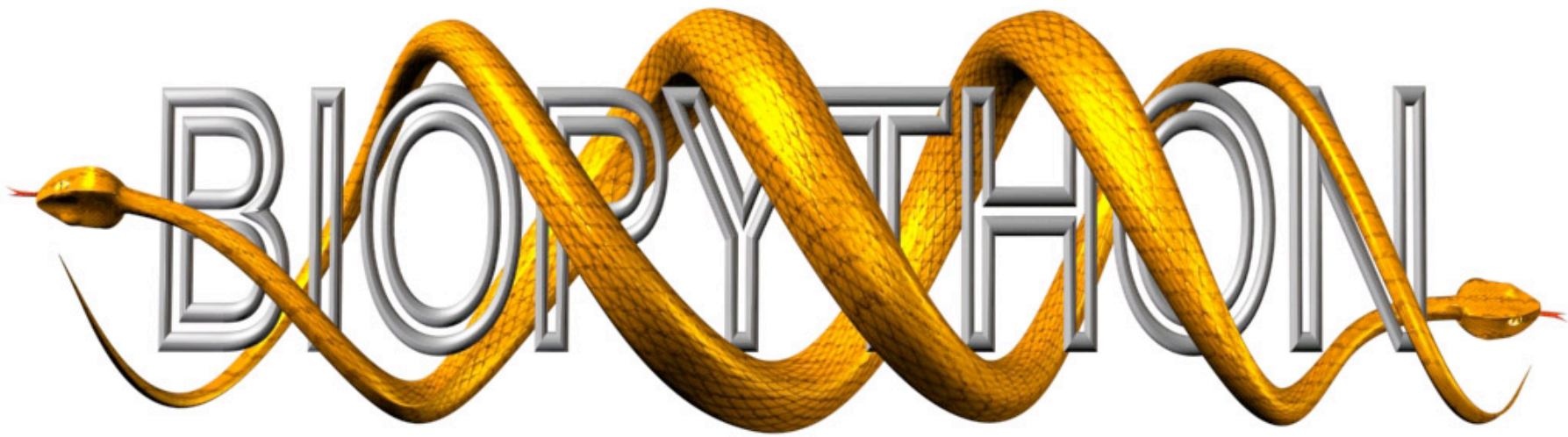
- Other Biopython contributors & developers!
- Open Bioinformatics Foundation (OBF) supports Biopython (and BioPerl etc)
- Society for General Microbiology (SGM) for my travel costs
- My Biopython work supported by:
  - EPSRC funded PhD (MOAC DTC, University of Warwick, UK)
  - SCRI (Scottish Crop Research Institute), who also paid my conference fees

What next?

- This afternoon's "Birds of a Feather" session:

  Biopython Tutorial and/or Hackathon

- Sign up to our mailing list?



- Homepage www.biopython.org

scri
living technology