Biopython Project Update 2013

Peter Cock & the Biopython Developers,
BOSC 2013, Berlin, Germany
Twitter: @pjacock & @biopython

Introduction

My Employer

- After PhD joined **Scottish Crop Research Institute**
- In 2011, SCRI (Dundee) & MLURI (Aberdeen) merged as **The James Hutton Institute**
- Government funded research institute
- I work mainly on the genomics of Plant Pathogens
- I use Biopython in my day to day work
- More about this in tomorrow's panel discussion, "*Strategies for Funding and Maintaining Open Source Software*"

Biopython

- Open source bioinformatics library for Python
- Sister project to:
  - BioPerl
  - BioRuby
  - BioJava
  - EMBOSS
  - etc (see OBF Project BOF meeting tonight)
- Long running!

Brief History of Biopython

- 1999 – Started by Andrew Dalke & Jeff Chang
- 2000 – First release, announcement publication
  - Chapman & Chang (2000). ACM SIGBIO Newsletter 20, 15-19
- 2001 – Biopython 1.00
- 2009 – Application note publication
  - Cock et al. (2009) DOI:10.1093/bioinformatics/btp163
- 2011 – Biopython 1.57 and 1.58
- 2012 – Biopython 1.59 and 1.60
- 2013 – Biopython 1.61 and 1.62 beta

Recap from last BOSC 2012

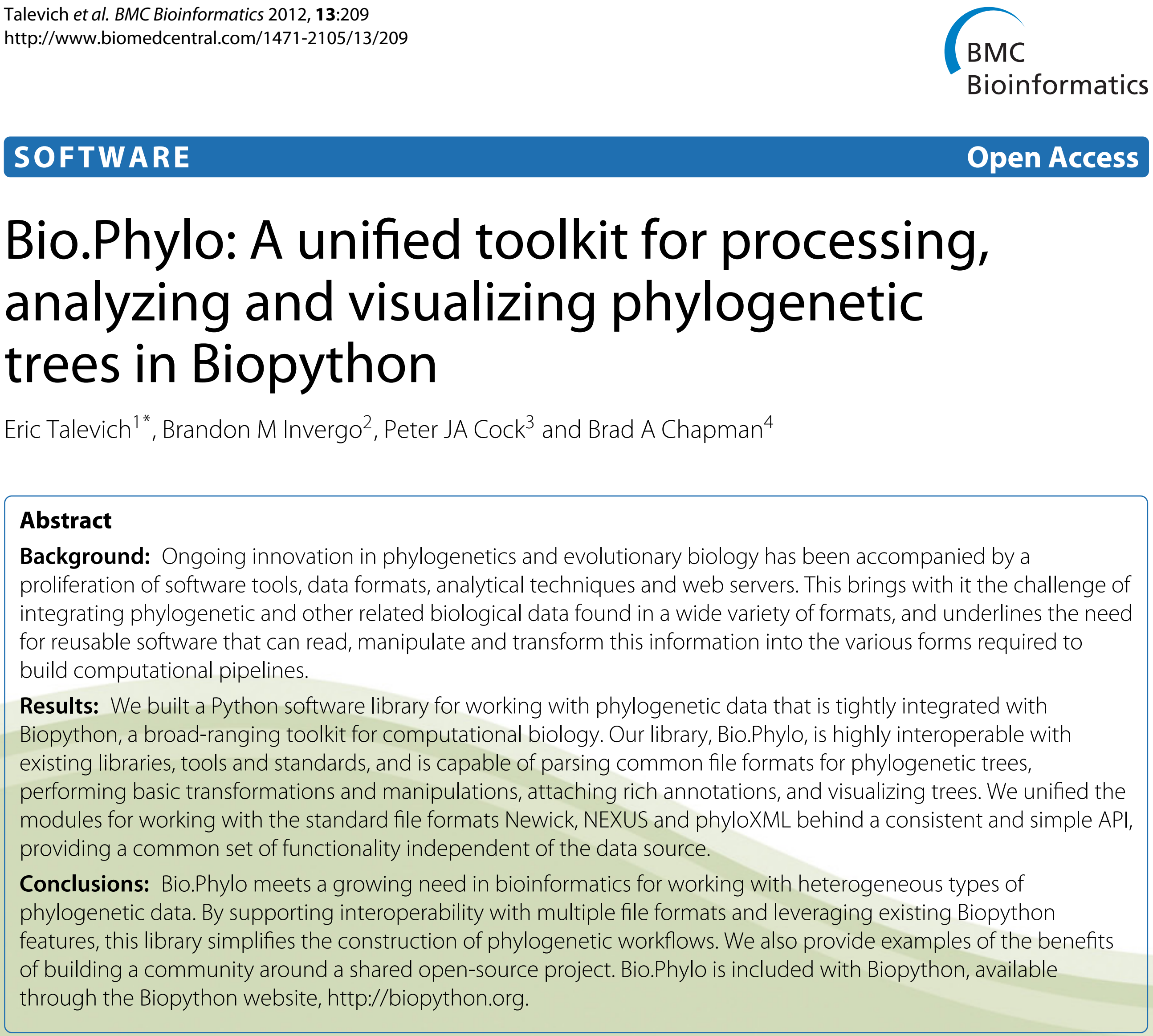
- Eric Talevich presented in Boston
- Biopython 1.58, 1.59 and 1.60
- Visualization enhancements for chromosome and genome diagrams, and phylogenetic trees
- More file format parsers
- BGZF compression
- Google Summer of Code students ...
- Bio.Phylo paper submitted and in review ...
- Biopython working nicely under PyPy 1.9 ...

The James Hutton Institute

Publications

Bio.Phylo paper published

- Talevich et al (2012) DOI:10.1186/1471-2105-13-209



Talevich *et al. BMC Bioinformatics* 2012, **13**:209
http://www.biomedcentral.com/1471-2105/13/209

BMC Bioinformatics

SOFTWARE Open Access

Bio.Phylo: A unified toolkit for processing, analyzing and visualizing phylogenetic trees in Biopython

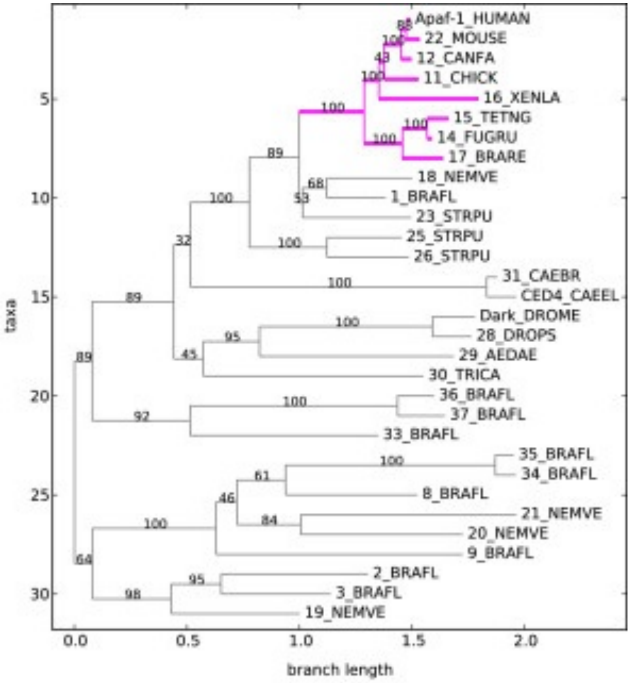
Eric Talevich[1*], Brandon M Invergo[2], Peter JA Cock[3] and Brad A Chapman[4]

Abstract

Background: Ongoing innovation in phylogenetics and evolutionary biology has been accompanied by a proliferation of software tools, data formats, analytical techniques and web servers. This brings with it the challenge of integrating phylogenetic and other related biological data found in a wide variety of formats, and underlines the need for reusable software that can read, manipulate and transform this information into the various forms required to build computational pipelines.

Results: We built a Python software library for working with phylogenetic data that is tightly integrated with Biopython, a broad-ranging toolkit for computational biology. Our library, Bio.Phylo, is highly interoperable with existing libraries, tools and standards, and is capable of parsing common file formats for phylogenetic trees, performing basic transformations and manipulations, attaching rich annotations, and visualizing trees. We unified the modules for working with the standard file formats Newick, NEXUS and phyloXML behind a consistent and simple API, providing a common set of functionality independent of the data source.

Conclusions: Bio.Phylo meets a growing need in bioinformatics for working with heterogeneous types of phylogenetic data. By supporting interoperability with multiple file formats and leveraging existing Biopython features, this library simplifies the construction of phylogenetic workflows. We also provide examples of the benefits of building a community around a shared open-source project. Bio.Phylo is included with Biopython, available through the Biopython website, http://biopython.org.



```
tree.root_at_midpoint()
tree.ladderize(reverse=True)
vertebrata = tree.common_ancestor(
            "Apaf-1_HUMAN", "15_TETNG")
vertebrata.color = "fuchsia"
vertebrata.width = 3
tree.root.color = "gray"
Phylo.draw(tree)
```

The James Hutton Institute

Google Summer of Code (GSoC)

Google Summer of Code 2012

- Two students under OBF
- Lenna Peterson
  - Genomic Variant Toolkit for Biopython
  - Mentors: Brad Chapman & James Casbon
  - http://arklenna.tumblr.com/tagged/gsoc2012
- Wibowo Arindrarto
  - Bio.SearchIO – pairwise sequence search files input/output, e.g. BLAST, HMMER
  - Mentor: Peter Cock
  - http://bow.web.id/blog/tag/gsoc/
- Both completed their projects & still contributing

Google Summer of Code 2013

- Two students under NESCent
- Zheng Ruan
  - Codon alignment and analysis
  - Mentors: Eric Talevich & Peter Cock
  - http://zr1991.blogspot.de/
- Yanbo Ye
  - Bio.Phylo: filling in the gaps
  - Mentor: Eric Talevich
  - http://blog.yeyanbo.com/tag/gsoc.html

Releases since BOSC 2012

Biopython 1.61

- Major refresh of sequence motif handling code
- Bio.SearchIO GSoC work as an experimental module
- Contributors:
  - Brandon Invergo
  - Bryan Lunt (*)
  - Christian Brueffer (*)
  - David Cain
  - Eric Talevich
  - Grace Yeo (*)
  - Jeffrey Chang
  - Jingping Li (*)
  - Kai Blin (*)
  - Leighton Pritchard
  - Lenna Peterson
  - Lucas Sinclair (*)
  - Michiel de Hoon
  - Nick Semenkovich (*)
  - Peter Cock
  - Robert Ernst (*)
  - Tiago Antao
  - Wibowo 'Bow' Arindrarto

Biopython 1.62

- Beta released
- Final release after BOSC/ISMB/ECCB
- Warning on translating partial codons
  - Explicit is better than implicit!
- Parsers for GAF, GPA and GPI from UniProt-GOA
- Reworked feature location object model
  - Cleaner handling of multi-region locations
  - Linked to GTF/GFF3 parsing and other plans
- Official Python 3 support ...

The James Hutton Institute

Please test this beta!

Biopython 1.62

- Contributors (as of the beta release):
  - Alexander Campbell (*)
  - Andrea Rizzi (*)
  - Anthony Mathelier (*)
  - Ben Morris (*)
  - Brad Chapman
  - Christian Brueffer
  - David Arenillas (*)
  - David Martin (*)
  - Eric Talevich
  - Iddo Friedberg
  - Jian-Long Huang (*)
  - Joao Rodrigues
  - Kai Blin
  - Michiel de Hoon
  - Nate Sutton (*)
  - Peter Cock
  - Petra Kubincová (*)
  - Phillip Garland
  - Saket Choudhary (*)
  - Tiago Antao
  - Wibowo 'Bow' Arindrarto
  - Xabier Bello (*)

Python 3

Python 2 and 3

- Python 2.7 is the final release of Python 2
- Python 3 is similar but different to Python 2
- Most Python 2 code needs updating to run
- Big difference is Python 3 uses unicode for strings
  - We need to be explicit about bytes vs unicode in many of our parsers
  - Text file IO defaults to unicode, which is slower
- Most Python libraries are gradually being updated

Python 3 strategy

- We’ve been testing under Python 3 for over a year
- Biopython 1.62 will officially support Python 3.3
- Current strategy:
  - Develop under Python 2
  - Installation under Python 3 uses 2to3 converter
  - Test under Python 3

Cross Platform Testing

- Nightly tests with BuildBot
  - Tests run on volunteer machines
  - Covers multiple OS and Python combinations
    - More volunteer machines welcome, especially 64 bit Windows
  - Server runs on OBF funded Amazon server
- Continuous integration with TravisCI
  - Covers a range of languages using VMs
  - Free service for Open Source projects
  - Runs tests when code on GitHub updated
  - Runs tests for GitHub pull requests (Nice!)

Cross Platform Testing

Python/OS	Linux 32 bit	Linux 64 bit	Mac OS X 64 bit	Windows 32 bit
C Python 2.5	BB + Travis	BB	BB	BB
C Python 2.6	BB + Travis	BB	BB	BB
C Python 2.7	BB + Travis	BB		BB
C Python 3.1	BB	BB	BB	BB
C Python 3.2	BB	BB	BB	BB
C Python 3.3	BB + Travis	BB		BB
PyPy 1.9	Travis		BB	BB
PyPy 2.0				BB
Jython 2.5		BB		BB
Jython 2.7b		BB		BB

The James Hutton Institute

This test matrix is quite big!

Cross Platform Testing

Python/OS	Linux 32 bit	Linux 64 bit	Mac OS X 64 bit	Windows 32 bit
~~C Python 2.5~~	~~BB + Travis~~	~~BB~~	~~BB~~	~~BB~~
C Python 2.6	BB + Travis	BB	BB	BB
C Python 2.7	BB + Travis	BB		BB
C Python 3.1	BB	BB	BB	BB
C Python 3.2	BB	BB	BB	BB
C Python 3.3	BB + Travis	BB		BB
PyPy 1.9	Travis		BB	BB
PyPy 2.0				BB
~~Jython 2.5~~		~~BB~~		~~BB~~
Jython 2.7b		BB		BB

Dropping Python 2.5 support

Also means Jython 2.5

Cross Platform Testing

Python/OS	Linux 32 bit	Linux 64 bit	Mac OS X 64 bit	Windows 32 bit
~~C Python 2.5~~	~~BB + Travis~~	~~BB~~	~~BB~~	~~BB~~
C Python 2.6	BB + Travis	BB	BB	BB
C Python 2.7	BB + Travis	BB		BB
~~C Python 3.1~~	~~BB~~	~~BB~~	~~BB~~	~~BB~~
~~C Python 3.2~~	~~BB~~	~~BB~~	~~BB~~	~~BB~~
C Python 3.3	BB + Travis	BB		BB
PyPy 1.9	Travis		BB	BB
PyPy 2.0				BB
~~Jython 2.5~~		~~BB~~		~~BB~~
Jython 2.7b		BB		BB

Have been useful in Python 3 testing, but won't support

Cross Platform Testing Plan

Python/OS	Linux 32 bit	Linux 64 bit	Mac OS X 64 bit	Windows 32 bit	Windows 64 bit
C Python 2.6	BB + Travis	BB	BB	BB	
C Python 2.7	BB + Travis	BB		BB	BB
C Python 3.3	BB + Travis	BB		BB	
PyPy 1.9	Travis		BB	BB	
PyPy 2.0				BB	
PyPy 2.1b					
Jython 2.7b		BB		BB	

Python 2.7 variants

- Target Python 2.6, 2.7 and 3.3 (or later)
- Volunteer machines needed, especially 64 bit Windows

Python 3 strategy

- Current strategy:
  - Develop under Python 2
  - Installation under Python 3 uses 2to3 converter
  - Test under Python 3

- Future strategy:
  - Target Python 2.6, 2.7 and 3.3 (or later)
  - Start writing code which works on *both* at same time
  - Continue to use 2to3 on a case-by-case basis during transition period, or for problem cases

Closing Remarks

Stability versus Flexibility

- We aim for rigorous cross-platform testing
- We value backwards compatibility in core code

- Flexibility through modularity?
  - BioRuby’s Gems http://gems.bioruby.org
  - BioPerl sub-packages on CPAN
  - Can/should we move towards this with PyPI?
  - Would this encourage more contributors?

- We’re trying ‘beta level’ experimental modules within the monolithic Biopython distribution, e.g. SearchIO

Biopython Support: Resources

- OBF hosted website, mailing lists, bug tracker, etc
- GitHub hosted repository
- TravisCI hosted continuous integration testing
- Personal and institute BuildSlave machines for testing

Thank you all!

Biopython Support: People

- Google supports summer students via GSoC
- Some of the developers do contribute on work time
- However, Biopython is mainly volunteer funded
- Please help out, e.g.
  - Feedback
  - Bug reports
  - Documentation improvements
  - Review code
  - More unit tests
  - Enhancements or new code
  - …

Thank you all!