Advanced Topics: Biopython

Day One - Introduction

Peter J. A. Cock

The James Hutton Institute, Invergowrie, Dundee, DD2 5DA, Scotland, UK

Monday 23rd January 2012,
Workshop on Genomics, Český Krumlov, Czech Republic

Talk Outline

Why programming?

- Bioinformaticians *need* to be able to script and program
- (Arguably before long so will *most* biologists)

A brief CV and how it relates to Programming

School and pre-university:

- BASIC, specifically QBASIC

Undergraduate Masters in Mathematics & Physics (MPhys):

- Fortan 90 and Pascal for numerical computation on Unix
- HTML for webpages, $\LaTeX$ for scientific reports

Worked in IT for a few years

- C programming on DOS, Visual Basic on Windows, SQL
- Importance of documentation and testing!

Interdisciplinary Masters (MSc), and Bioinformatics PhD

- Python including Biopython, and a little MatLab

Bioinformatics Postdoc (at SCRI/JHI)

- More Python and Biopython

Right language for the task?

- Scripting: Automating tasks, gluing tools together
  - Usually written in Perl, shell script, Python, . . .
- Tools: Specialist programs to do one job
  - Often written in a compiled language like C for speed
- Webtools: Often a front end to existing tools
  - Often written in an interpreted language like PHP, Perl, Ruby or Python

Python is flexible enough to do all of this.

Python eco-system

- Link to C or Fortan code (for speed)
- Numerics: NumPy and SciPy
- Graphics: MatPlotLib (pylab), ReportLab, . . .
- Databases: SQLAlchemy, . . .
- Websites: Danjo, TurboGears
- Biology: Biopython, Galaxy, . . .

What is 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 more details

Why Biopython?

Python	Perl	Ruby	Java
Biopython	BioPerl	BioRuby	BioJava

- All the Bio* OBF projects are now mature and capable. . .
- Which programming language are you comfortable in?
- Which language/library do your group/colleagues use?
- Do you have any specific needs to narrow the choice?

Why did I choose Biopython? Because I chose Python

- I knew several programming languages already
- There were no other programmers in my research group
- I didn't know Perl, Java or Ruby:
  - I'd looked at Perl and didn't like it
  - Java seemed too heavy for scripting
  - I don't recall being aware of Ruby as an option
- I'd been introduced to Python during my MSc and *liked it!*

Zen of Python

- Beautiful is better than ugly
- Explicit is better than implicit
  ...
- Readability counts
  ...
- Special cases aren't special enough to break the rules
- Although practicality beats purity
- Errors should never pass silently
  ...
- If the implementation is hard to explain, it's a bad idea
- If the implementation is easy to explain, it may be a good idea
  ...

Full list at `http://www.python.org/dev/peps/pep-0020/` or `import this`

Selected Biopython examples

- Seq objects
- FASTA files and `Bio.SeqIO`
- FASTQ files and `Bio.SeqIO`

Seq object – like a Python string

```
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import generic_dna
>>> dna = Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC",
...           generic_dna)
>>> print dna, dna.alphabet
GATCGATGGGCCTATATAGGATCGAAAATCGC DNAAlphabet()
>>> len(dna)
32
>>> dna.count('C')
6
>>> dna.find("TATAT")
12
>>> print dna.lower()
gatcgatgggcctatataggatcgaaaatcgc
```

Seq object – Biological methods

```
>>> print dna, dna.alphabet
GATCGATGGGCCTATATAGGATCGAAAATCGC DNAAlphabet()

>>> rc = dna.reverse_complement()
>>> print rc, rc.alphabet
GCGATTTTCGATCCTATATAGGCCCATCGATC DNAAlphabet()

>>> rna = dna.transcribe()
>>> print rna, rna.alphabet
GAUCGAUGGGCCUAUAUAGGAUCGAAAAUCGC RNAAlphabet()

>>> protein = rna.translate()
>>> print protein, protein.alphabet
DRWAYIGSKI ExtendedIUPACProtein()
```

FASTA files and Bio.SeqIO

```
>FL3BO7415JACDX
TTAATTTTATTTTGTCGGCTAAAGAGATTTTTAGCTAAACGTTCAATTGCTTTAGCTGAA
GTACGAGCAGATACTCCAATCGCAATTGTTTCTTCATTTAAAATTAGCTCGTCGCCACCT
TCAATTGGAAATTTATAATCACGATCTAACCAGATTGGTACATTATGTTTTGCAAATCTT
GGATGATATTTAATGATGTACTCCATGAATAATGATTCACGTCTACGCGCTGGTTCTCTC
ATCTTATTTATCGTTAAGCCA
>FL3BO7415I7AFR
CATTAACTAA...
```

```
#Print record identifiers, length and first ten bases
from Bio import SeqIO
for rec in SeqIO.parse("phage.fasta", "fasta"):
    print rec.id, len(rec.seq), rec.seq[:10]+"..."
```

```
FL3BO7415JACDX 261 TTAATTTTAT...
FL3BO7415I7AFR 267 CATTAACTAA...
FL3BO7415JCAY5 136 TTTCTTTTCT...
FL3BO7415JB41R 208 CTCTTTTATG...
FL3BO7415I6HKB 268 GGTATTTGAA...
FL3BO7415I63UC 219 AACATGTGAG...
...
```

FASTQ files and Bio.SeqIO

```
@FL3BO7415JACDX
TTAATTTTATTTTGTCGGCTAAAGAGATTTTTAGCTAAACGTTCAATTGCTTTAGCTGAAGTACGAGCAGATACTCCAATCGCAATTGTTTCTTC
ATTTAAAATTAGCTCGTCGCCACCTTCAATTGGAAATTTATAATCACGATCTAACCAGATTGGTACATTATGTTTTGCAAATCTTGGATGATATT
TAATGATGTACTCCATGAATAATGATTCACGTCTACGCGCTGGTTCTCTCATCTTATTTATCGTTAAGCCA
+
BBBB2262=1111FFGGGHHHHIIIIIIIIIIIIIIIIIIIIIIIFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFGGGGFFFFFFFFFFFFFFFFFGB
BBCFFFFFFFFFFFFFFFFFFFFFFFFFGGGGGGGIIIIIIIGGGIIIGGGIIGGGG@AAAAA?===@@@???
@FL3BO7415I7AFR
CATTAACTAA...
```

```python
#Changed filename and format from "fasta" to "fastq"
from Bio import SeqIO
for rec in SeqIO.parse("phage.fastq", "fastq"):
    print rec.id, len(rec.seq), rec.seq[:10]+"..."
```

```
FL3BO7415JACDX 261 TTAATTTTAT...
FL3BO7415I7AFR 267 CATTAACTAA...
FL3BO7415JCAY5 136 TTTCTTTTCT...
FL3BO7415JB41R 208 CTCTTTTATG...
FL3BO7415I6HKB 268 GGTATTTGAA...
FL3BO7415I63UC 219 AACATGTGAG...
...
```

A personalised Biopython history

- Biopython early history (first five years)
- Biopython when I got involved (about five years ago)
- Biopython recent history
- Retrospective

Biopython early history

- 1999: Started by Jeff Chang & Andrew Dalke
- 2000: Biopython 0.90, first release
- 2001: Biopython 1.00, “semi-complete”
- 2002: Biopython 1.10, “semi-stable”
- 2003: Biopython 1.20, 1.21, 1.22 and 1.23,
  - Jeff Chang hands over to Brad Chapman as project leader
  - Logo created by Henrik Vestergaard and Thomas Hamelryck

Biopython late 2004 – Lucky timing for me?

- Feb 2004: Biopython 1.24 release
- May 2004: Biopython 1.30 release
- Sept 2004: I started my PhD, began using Biopython
- Oct 2004: Spam on mailing lists becoming a big problem
- Dec 2004: Iddo Friedberg assumes project leadership,

Email thread: [Biopython-dev] To the core developers...

James Casbon: Is biopython dead?
Iddo Friedberg: No but it sure smells funny. I apologize for that, ... Brad, Jeff: if you guys are busy, I can take over for a while.

- Feb 2005: Biopython 1.40 *beta* released
  (I had made some very minor contributions)
- March 2005: Most of spam on list archives removed
  Mailing lists made subscriber only (to reduce spam)

Biopython during my PhD

- Sept 2004: I started my PhD, began using Biopython
- Oct 2005: Biopython 1.41 released
- July 2006: Biopython 1.42 released (with my GenBank code)
- Mar 2007: Biopython 1.43 released (with my SeqIO code)
- Oct 2007: Biopython 1.44 released
- Mar 2008: Biopython 1.45 released
- May 2008: I start my new job at SCRI
- June 2008: Biopython 1.46 not released
  (due to my adding a last minute bug).
- July 2008: Biopython 1.47 released (with AlignIO code)
- August 2008: I submitted my PhD thesis

Now a steady release process, every three or four months

How has Biopython has changed?

- I’d like to think the documentation and sequence basics is much easier for beginners now than five year ago.
- I’d also like to think the project has matured:
  - We have a deprecation policy for phasing out old code
  - New code must have unit tests and documentation
- Switching to a distributed version control system (git) has made it much easier for anyone to develop experimental new code, and do so in public (on the github.com website).

Looking back on my involvement

- The timing was fortunate: At the start of my PhD, Biopython development had stalled - most of the key contributors were post-docs or more senior, and could not spare much time.
- As a PhD student, I had the time and the flexibility to work on Biopython code to support my own research needs.
- I'm lucky to have been able to continue to spend time on Biopython while working

Getting involved (in Biopython)

You are all potential contributors! First steps:

- Sign up to the mailing list(s)
- Ask questions on the mailing list
  (please also ask me in person during this workshop)
- File bug reports if you find problems in the code
- Follow @Biopython on Twitter ;)

Getting involved

Later, as you get to know Biopython better, try to:

- Answer questions on the mailing list
- Share solutions to problems you've found
  (e.g. on your blog, or as a cookbook entry on our wiki)
- Suggest things for the documentation

Getting involved (continued)

Once you feel more confident:

- Suggest bug fixes (patches or git branches)
- Write documentation
- Write unit tests (we always need more)
- Consider writing additional parsers, or modules – ideally pick things that are important in your research (this makes justifying it easier to your boss/supervisor).

Getting involved in Open Source in general

- Most of that advice to open source projects in general
- I’ve made small contributions to many projects
- I believe in open source for science and in general
  - For reproducible results, must share data *and* code

Personal Acknowledgements

- MOAC Doctoral Training Centre, University of Warwick, UK
- EPSRC (UK) for MSc and PhD funding
- The James Hutton Institute, previously known as the Scottish Crop Research Institute, my supportive employer
- *Workshop on Genomics* organisers and participants

Project Acknowledgements

- Open Bioinformatics Foundation (OBF)
  (Non-profit which looks after Bio* projects)
- BioTeam Inc.
  (Company that hosts the OBF servers)
- GitHub Inc.
  (Repository hosting)
- Google Summer of Code
  (Students funded in last three years)
- The many individuals who have contributed over the years

Next steps

- I want people to ask questions
- I do have more slides and examples planned,
- But first, some quick checks...

Next steps

- Have you all got Python 2.x installed?
- Have you all got a recent Biopython installed?
- Have you all found the Biopython website and wiki?
  `http://biopython.org`
- Have you all found the Biopython Tutorial?
  (online or local copy fine)
- Are you familiar with the `help(...)` command in Python?

Next steps

- Have you all got a recent NumPy installed?
- Have you all got a recent ReportLab installed?
- Do you have git installed?
- Do you know how to use git?
- Are you interested in Python 3?
- Are you interested in Jython or PyPy?