Biopython project update

Tiago Antao[1], Peter Cock[2]

In this talk we present the current status of the Biopython project. We start by giving a short overview of Biopython (presenting existing functionality) and useful software libraries for computational biology in the Python development 'ecology' (from plotting libraries capable of producing publication quality figures to numerical libraries, among others). We then focus on features developed since BOSC 2007, future plans for the project and present example usages of the new population genetics module.

The latest Biopython release is 1.45 made available on 22 March 2008.

Some of the new features are:

1. A new population genetics module including support for coalescent simulation, selection detection and the GenePop file format. The new module relies on existing open source external software (e.g., the open source Simcoal2 for coalescent simulation which can take advantage of multiple core CPUs for computationally intensive tasks).

2. Improved documentation.

3. Deprecation of many modules which were either obsolete or had been superseded by other code.

4. Plus many bugs were fixed, including updates for evolving file formats.

Since the Biopython 1.45 release, further work is planned to extend the Population Genetics module (e.g., with a statistics component). A new sequence alignment module is also being implemented with a uniform API for reading and writing various alignment files, based on the approach of the Bio.SeqIO module added last year for working with sequences. Work to improve Biopython's BioSQL support is also ongoing.

Time permitting, the talk will also show usage examples of the new population genetics module. The focus will be put not only on the population genetics side, but also on strategies to easily use all available computational power on new multiple core computers. This is useful for users of most scripting languages as most language interpreter implementations impose stern limits on multi-threaded programming efficiency, a topic that becomes important when using computational biology code that is CPU intensive. We will take this opportunity to discuss strategies to overcome those interpreter limitations.

Web site: http://www.biopython.org

License: Biopython License Agreement (an MIT style license)
http://www.biopython.org/DIST/LICENSE

---

1   Liverpool School of Tropical Medicine, UK
2   University of Warwick, UK