SeqHoundRWeb.py: interface to a comprehensive online bioinformatics resource

Peter St. Onge
University of Toronto
pete@{seul.org|economics.utoronto.ca}

Paul Osman
paul@eval.ca

Abstract

In the post-genomic era, getting useful answers to challenging biological questions often demands significant expertise and resources not only to acquire the requisite biological data but also to manage it. The storage required to maintain a workable genomic or proteomic database is usually out of reach for most biologists. Some toolsets already exist to facilitate some aspects of data analysis, and others for access to particular data stores (e.g., NCBI Toolkit), but there is a substantial learning curve to these tools and installation is often non-trival. SeqHoundRWeb.py grew out of a common frustration in bioinformatics—the initiate bioinformaticist often has substantial biological knowledge, but little experience in computing; Python is often held up as a good first scripting language to learn, and in our experience new users can be productive fairly rapidly.

Introduction

The discovery of DNA by Watson and Crick marked the beginnings of massive upheaval in biology, and ultimately, in the ways biologists work. Research today, in the so-called post-genomic era, has embraced computing technology as never before, with repercussions affecting all areas of biology[15].

One of the greatest hurdles a novice bioinformaticist must face is the learning curve when learning an approach to biology that does not involve the use of any of the classical or well-known laboratory techniques.

Perl is probably the most commonly used bioinformatics language currently[13], and has substantial and rich object-oriented facilities to deal with biological data[2]. While Perl is highly versatile and effective in the hands of experienced bioinformaticists[14], my experience shows that initiates to programming through academic bioinformatics courses often have considerable difficulty understanding the breadth of Perl approaches and idioms sufficiently for it to be useful.

Like Perl, other languages have seen specialized facilities to handle biological information develop considerably. Java[1], Ruby[4], Python[3], amongst others, all have have been used successfully in research projects. In particular, use of Python is becoming increasingly commonplace in research[12].

A second issue is the ability to obtain and effectively exploit data in order to test the research hypotheses of interest. Fortunately, there are many research sites providing substantial data for research use, including the US National Center for Biotechnology Information[7] (NCBI), the Gene Ontology Consortium[5] (GO) and the European Bioinformatics Institute[8] (EBI). Each of these sites host a considerable amount of data, in both size and breadth, typically database table dumps allowing others to reconstitute and further develop the data for individual research needs. Although the data files are normally well under a gigabyte in size (compressed), typical processing requires some skill even for simple parsing and data work up, particularly because of the file sizes involved.

Not surprisingly, most research questions tend to be more complex and require more robust approaches to managing data, such as relational databases (e.g. MySQL and PostGreSQL) which in turn often mean having substantial hardware and some system administration skills.

Other types of data processing, such as genome-level comparisons between species through Basic Local Alignment Search Tool[10] (BLAST) can be highly CPU intensive for hours or even days depending on the size of the genomes being searched and the underlying hardware. As this data updates on an ongoing basis, the need to rebuild result sets dictates the availability of large quantities of substantial computing power.

One project to assemble much of the data from these various sources and carry out many of the more demanding data process steps was SeqHound[11], merging biological sequence (genomic and proteomic), taxonomy, annotation and 3-D structure within an object-oriented database management system and exposed via a web-based API. Although most of SeqHound is F/OSS, hosting it locally locally would be difficult for most researchers without substantial hardware and technical expertise, as the current (as of Oct '04) recommendation is to have a system able to store some 650 GB of data[9].

In order to make access to SeqHound simpler for novice bioinformaticists, my approach was to trade off some speed for flexibility, and build a Python wrapper around SeqHound's HTTP API, exposing much of the rich data provided by SeqHound to the ease-of-use of Python and its language features.

Overview

The primary prerequisites for SeqHoundRWeb.py are the urllib and os modules, so this means that SeqHoundRWeb.py should be able to work on any platform supported by Python. Installation of the package will be via the typical Python installation techniques, and we expect that it will be made available through the normal distribution channels soon. Until then, the code can be imported into Python, as shown below as long as the location of the SeqHoundRWeb.py file is provided—novice Python users can take full advantage of the functions provided in this module without the need for root or Administrator-level access!

Table 1 shows a straightforward example of retrieval of a GenInfo (GI) identifier given a particular accession number for a hypothetical protein for a species of rat (specifically, the Norway Rat, *Rattus norvegicus*).

Acknowledgements

This project came about thanks to a number of people: Alexander Ignachenko, Shaun Ghanny, Robin Haw, Henry Ling, Bianca Tong, Kayu Chin, Thomas Kislinger, and Ata Ghavidel all provided constructive criticism and support, for

```
import os
import SeqHoundRWeb

# Set the proper URL for seqhound
os.environ["SEQHOUNDSITE"] = "http://seqhound.blueprint.org"

accs = [
   "CAA28783",
   "CAA28784",
   "CAA28786"
]

for acc in accs:
   result = SeqHoundRWeb.SeqHoundFindAcc([acc])
   if result[0] == ’SEQHOUND_OK’: # found it
       print result[1]
```

Table 1: Simple SeqHoundRWeb Example - Rat

which I remain grateful. Katerina Michalickova, Michel Dumontier and others from the Hogue Lab at Mount Sinai Hospital for creating SeqHound and exposing its functionality via HTTP. An initial proof of concept, which became the basis for this project, was built in the Emili Lab at the University of Toronto, and I would like to thank the Department of Economics for allowing me the opportunity to continue working on it as part of my professional responsibilities. Thanks also to Alex Brotman, Ales Hvezda and others from the SEUL Project for their keen eyes in finding mistakes in the text. Any errors or omissions are, of course, my own[6].

References

[1] BioJava web site. `http://www.biojava.org/`.

[2] BioPerl web site. `http://bio.perl.org/`.

[3] BioPython web site. `http://www.biopython.org/`.

[4] BioRuby web site. `http://www.bioruby.org/`.

[5] Gene Ontology Consortium FTP site. `http://archive.godatabase.org/latest-full/`.

[6] It’s All Pete’s Fault website. `http://www.itsallpetesfault.org/`.

[7] NCBI FTP site. `http://www.ncbi.nlm.nih.gov/Ftp/`.

[8] Catherine Brooksbank, Evelyn Camon, Midori A. Harris, Michele Magrane, Maria Jesus Martin, Nicola Mulder, Claire O’Donovan, Helen Parkinson, Mary Ann Tuli, Rolf Apweiler, Ewan Birney, Alvis Brazma, Kim Henrick, Rodrigo Lopez, Guenter Stoesser, Peter Stoehr, and Graham Cameron. The European Bioinformatics Institute’s data resources. *Nucl. Acids Res.*, 31(1):43–50, 2003.

[9] Ian Donaldson, Katerina Michalickova, Hao Lieu, Renan Cavero, Michel Dumontier, Doron Betel, Ruth Isserlin, Marc Dumontier, Michael Matan, Rong Yao, Zhe Wang, Victor Gu, and Elizabeth Burgess. The SeqHound Manual, Release 3.01, October 2004.

[10] Mark Yandell Ian Korf and Joseph Bedell. *BLAST*. O'Reilly, 2003.

[11] Katerina Michalickova, Gary Bader, Michel Dumontier, Hao Lieu, Doron Betel, Ruth Isserlin, and Christopher Hogue. Seqhound: biological sequence and structure database as a platform for bioinformatics research. *BMC Bioinformatics*, 3(1):32, 2002.

[12] J. Daniel Navarro, Vidya Niranjan, Suraj Peri, Chandra Kiran Jonnalagadda, and Akhilesh Pandey. From biological databases to platforms for biomedical discovery. *Trends in Biotechnology*, 21(6):263–268, 2003.

[13] James Tisdall. *Beginning Perl for Bioinformatics*. O'Reilly, 2001.

[14] James D. Tisdall. *Mastering Perl for Bioinformatics*. O'Reilly, 2003.

[15] Johnathan D. Wren. Engineering in genomics. *IEEE Engineering in Medicine and Biology Magazine*, pages 87–98, March/April 2004.

Proceedings of the
Linux Symposium

Volume Two

July 20nd–23th, 2005
Ottawa, Ontario
Canada

Conference Organizers

Review Committee

Proceedings Formatting Team