PROGRAMMING LANGUAGES AND THE BIOLOGICAL SCIENCES

James W. McGuffee, Ph.D.
St. Edward's University
Austin, TX 78704
512 448-8465
jameswm@stedwards.edu

ABSTRACT

This paper presents an overview of the scripting, markup, and general purpose programming languages used for scientific research and discovery in the biological sciences. This material may be useful for inclusion in a programming languages class and as a starting point for undergraduate interdisciplinary research and programming projects.

INTRODUCTION

This paper will present a brief overview of the various programming language projects that are ongoing in the biological sciences. This paper is not an attempt to catalog every programming project in the biological sciences. Rather, the criteria for inclusion in this survey is that the project is primarily concerned with creating tools to be used by programmers that create software in the biological sciences.

Throughout this paper there are several references to the O|B|F. O|B|F is the abbreviation for the Open Bioinformatics Foundation, a working coalition of several of the bioinformatics programming language projects. Specifically, the O|B|F grew out of the following volunteer projects: BioPerl, BioJava, and BioPython. The primary activity of the O|B|F is to support the Bioinformatics Open Source Conference (BOSC) [16].

SCRIPTING LANGUAGES

Currently, the most popular use of computer programming languages in the biological sciences is in the field of bioinformatics. A frequent task needed in bioinformatics is to find certain patterns in data files. The problem is that these data files often contain information in various formats. A solution is to use regular expressions to specify these patterns so that the information can be mined from the data files. Scripting languages are often used in bioinformatics due to their ability to work with regular expressions [4].

Perl

Bioperl is arguably the most successful project that is part of the O|B|F. Bioperl code is an extensive library of core modules written in Perl to support the processing, manipulating, and managing of biological information in the form of sequences [19]. One of the reasons that the Perl programming language was chosen was that it had already gained popularity in the bioinformatics community for its support of text processing and pattern matching task. The Bioperl project attempts to emulate the object-oriented programming paradigm through the use of Perl modules and by adhering to three design principles. The first principle is to separate the interface from the implementation. The second principle is to provide a base framework for the respective operation by generalizing common routines into a single module. The third and final principle is to use the Factory and Strategy patterns as defined by Erich Gamma [20]. For more information, go to http://www.bioperl.org/.

Python

In general, high-level scripting languages are popular language choices for researchers in the field of bioinformatics. In addition to its abilities as a scripting language, Python has the added support of advanced numerical capabilities through the Scientific Tools for Python (SciPy) project [6,14].

The Biopython project was created in 1999 and is modeled after the successful Bioperl project [3]. Most of the work of the Biopython project has focused on creating parsers for biological data and designing a useful interface to represent sequences. One of the unique features of the Biopython project is the use of a standard event-oriented parser design. For more information, go to http://biopython.org/.

PHP

Formerly known as GenePHP, the BioPHP project seeks to extend the PHP language so that it can be used to develop bioinformatics applications [10]. The main purpose of the BioPHP project is to encourage the use of PHP as a “glue” language to bind web-based bioinformatics applications and databases. Some of the tasks that BioPHP can currently do is read biological data in the GenBank, Swissprot, Fasta, and Clustal ALN formats and perform simple sequence analysis tasks. For more information, go to http://genephp.sourceforge.net/.

Ruby

BioRuby is part of the O|B|F and is primarily supported by the Human Genome Center at the University of Tokyo and the Bioinformatics Center at Kyoto University. One of the advantages of using Ruby for bioinformatics is that is has native support for object-oriented programming with a simple but powerful syntax [9].

In addition to the BioRuby project, several educators have advocated the desirability of introducing undergraduate students to the Ruby programming language [2]. For example, Daniel Lim introduced Ruby as a tool for bioinformatics in his programming languages class [11]. Not only was it an in-class success but several students continued the work as an independent project.

GENERAL PURPOSE LANGUAGES

Sometimes, researchers in the biological sciences will choose a programming language based on its general use and popularity. Below is a list of several projects that use traditional high-level general purpose programming languages.

C

The Laboratory of DNA Information Analysis at the University of Tokyo has produced an open source C library of the most commonly used clustering algorithms [7]. Clustering routines are used to analyze gene expression data. The library, as written, is callable from any program written in C or C++. Extensions have also been developed to allow these programs to be used in Perl and Python programs.

C++

Gianluca Della Vedova leads the Algorithms Library for Bioinformatics (ALiBio) project at the University of Milan-Bicocca. The stated goal of the project is to provide a library of fundamental, C++ implemented algorithms that will be used to develop applications in the bioinformatics field [21]. While most bioinformatics programming projects have focused on making tools that are easy to use, the ALiBio project values efficiency as its top priority. The goal is to provide tools to help produce highly optimized applications. In addition to the stated goal of efficiency, all libraries and algorithms that are included in ALiBio must have an extensive suite of regression tests and must be clearly documented.

The ALiBio project began in March 2002. The latest version (0.9) was released in January 2003. For more information, go to http://bioinformatics.org/ALiBio/.

Java

BioJava is an open source Java library and part of the O|B|F [17]. The BioJava project is primarily concerned with how to represent sequences. The major feature of the BioJava project is that two unique representation schemes for sequences have been defined in the Java programming language. The first scheme is the basic string-of-token representation and is used when annotation is unimportant in the analysis of the data. The second representation is the annotated sequence framework and is used when a fully annotated view of the sequence is required. More information about BioJava may be found at http://biojava.org/.

Squeak

One of the more disappointing aspects of examining programming languages and how they are being used in the biological sciences is to find a project that seems promising but is ultimately a disappointment. Squeak is an open source implementation of the object-oriented programming language Smalltalk. With its built in graphics and truly object-oriented paradigm, Squeak would seem an ideal language for the biological sciences. In fact, the bioSqueak homepage seems to make such promises [15]. However, there is no evidence that any work has actually been completed on the bioSqueak project.

FUNCTIONAL AND LOGIC LANGUAGES

The projects that use the languages in this category are as varied as the languages themselves.

Haskell

Haskell is a general purpose, purely functional programming language. Haskell has been used by Robert Giegerich's research group to implement dynamic programming algorithms, including those involved in RNA folding grammars [8]. The group claims that their work adds a significant amount of flexibility and versatility in the development of new dynamic programming algorithms.

Lisp

Formerly known as BioLingua, BioBike is an interactive, web-based programming environment that enables biologists to analyze biological systems by combining knowledge and data through direct end-user programming [12]. The goal of BioBike is to enable biologists to program directly by providing an environment that is more natural to trained biologists. The main BioBike language is BioLisp. BioLisp is simply common Lisp with added biological functionality [18].

Prolog

Written in SWI-Prolog, Biomedical Logic Programming (Blip) is a collection of logic programming modules intended primarily for bioinformatics and biomedical applications [13]. Blip allows users to program in a declarative way and facilitates both query-oriented and application-oriented programming. For more information, go to http://bioprolog.org/.

XML – A MARKUP LANGUAGE

While not technically a programming language, no discussion of biological oriented programming would be complete without a brief discussion of XML. When dealing with massive amounts of data in different formats, there must be a way to exchange this information from one application to another. XML is an extensible, universal format for structured data exchange and documents on the web [5]. Two of the most notable attempts to use XML as a framework in biology have been the Bioinformatics Sequence Markup Language (BSML) and the BIOpolymer Markup Language (BioML) [1].

CONCLUSIONS

This paper has presented a very broad overview of the various programming language projects in the biological sciences. One way to use this information is as instructional material in a computer programming languages course. By giving specific examples, students can start to understand the decisions involved in selecting a language for a specific application.

What this paper has not done is analyze the relative merits of using one programming language versus another. This is research that should be done. To that end, the author has created an on-going project called the Bio-Languages Research Laboratory. This virtual laboratory is intended to be an information center for on going projects that examine the uses of computer programming languages in the biological sciences. If you or your students are interested in learning more, please visit the "Bio-Languages Research Laboratory" at <faculty.stedwards.edu/jameswm/blrl/>.

REFERENCES

1. Achard, F., Vaysseix, G., Barillot, E., XML, bioinformatics and data integration, *Bioinformatics*, 17, (2), 115-125, 2001.
2. Baas, B., Ruby in the cs curriculum, *Journal of Computing Sciences in Colleges*, 17, (5), 95-103, 2002.
3. Chapman, B., Chang, J., Biopython tools for computational biology, *ACM SIGBIO Newsletter*, 20, (2), 15-19, 2000.
4. Cohen, J., Bioinformatics—an introduction for computer scientists, *ACM Computing Surveys*, 36, (2), 122-158, 2004.

5. Cohn, J., XML and genomic data, *ACM SIGBIO Newsletter*, 20, (3), 22-24, 2000.
6. de Hoon, M.J.L., Chapman, B., Friedberg, I., Bioinformatics and computational biology with biopython, *Genome Informatics*, 14, 298-299, 2003.
7. de Hoon, M.J.L., Imoto, S., Nolan, J., Miyano, S., Open source clustering software, *Bioinformatics*, 20, (9), 1453-1454, 2004.
8. Giegerich, R., Haskell in bioinformatics, <biowiki.org/BioHaskell>, 2006.
9. Goto, N., Nakao, M.C., Kawashima, S., Katayama, T., Kanehisa, M., BioRuby: open-source bioinformatics library, *Genome Informatics*, 14, 629-630, 2003.
10. Gregorio, S., The BioPHP project (formerly GenePHP), <genephp.sourceforge.net>, 2003.
11. Lim, D., A Ruby in the rough: using VHLLs in bioinformatics, *Journal of Computing Sciences in Colleges*, 21, (6), 108-116, 2006.
12. Massar, J.P., Travers, M., Elhai, J., Shrager, J., BioLingua: a programmable knowledge environment for biologists, *Bioinformatics*, 21, (2), 199-207, 2005.
13. Mungall, C., Blipkit: biomedical logic programming knowledge integration kit, <bioprolog.org>, 2005.
14. Oliphant, T., Scientific tools for Python, <www.scipy.org>, 2006.
15. O'Neel, B., bioSqueak home, <biosqueak.sourceforge.net>, 2002.
16. Open Bioinformatics Foundation, <www.open-bio.org>, 2006.
17. Pocock, M., Down, T., Hubbard, T., BioJava: open source components for bioinformatics, *ACM SIGBIO Newsletter*, 20, (2), 10-12, 2000.
18.Shrager, J., Massar, J.P., Travers, M., BioBike documentation index, <nostoc.stanford.edu/Docs/>, 2006.
19. Stajich, J., Birney, E., The bioperl project: motivation and usage, *ACM SIGBIO Newsletter*, 20, (2), 13-14, 2000.
20. Stajich, J., et. al., The bioperl toolkit: perl modules for the life sciences, *Genome Research*, 12, 1611-1618, 2002.
21. Vedova, G.D., Dondi, R., A library of efficient bioinformatics algorithms, *Applied Bioinformatics*, 2, (2), 117-121, 2003.