BIOINFORMATICS APPLICATIONS NOTE Vol. 29 no. 18 2013, pages 2355–2356
doi:10.1093/bioinformatics/btt398

Sequence analysis Advance Access publication July 9, 2013

BioSmalltalk: a pure object system and library for bioinformatics

Hernán F. Morales* and Guillermo Giovambattista

Instituto de Genética Veterinaria (IGEVET), CONICET La Plata–Facultad de Ciencias Veterinarias, Universidad Nacional de La Plata, La Plata B1900AVW, CC 296 Argentina

Associate Editor: Janet Kelso

ABSTRACT

Summary: We have developed BioSmalltalk, a new environment system for pure object-oriented bioinformatics programming. Adaptive end-user programming systems tend to become more important for discovering biological knowledge, as is demonstrated by the emergence of open-source programming toolkits for bioinformatics in the past years. Our software is intended to bridge the gap between bioscientists and rapid software prototyping while preserving the possibility of scaling to whole-system biology applications. BioSmalltalk performs better in terms of execution time and memory usage than Biopython and BioPerl for some classical situations.

Availability: BioSmalltalk is cross-platform and freely available (MIT license) through the Google Project Hosting at http://code.google.com/p/biosmalltalk

Contact: hernan.morales@gmail.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on January 12, 2013; revised on June 5, 2013; accepted on July 3, 2013

1 INTRODUCTION

We present a novel free/open source software (FOSS) platform for the development of bioinformatics software and applications. BioSmalltalk attempts to reconcile the current *de facto* scripting modalities of textual programming languages with the features of Smalltalk (Goldberg and Robson, 1983), which has a pure object dynamic programming environment.

BioSmalltalk provides similar functionality to other FOSS toolkits for bioinformatics, such as BioPerl (Stajich *et al.*, 2002), Biopython (Cock *et al.*, 2009) and BioJava (Holland *et al.*, 2008), based in industry-leading general-purpose textual programming languages. Precedent of bioinformatics tools exists in Smalltalk, but none of them has tried to provide a bioinformatics Application Programming Interface (API). MolTalk (Diemand and Scheib, 2004) was developed in StepTalk, a scripting environment, for doing structural bioinformatics. Also, a cross-platform Graphical User Interface (GUI) for protein sequence analysis was done in Smalltalk (Wishart *et al.*, 1997).

Object-orientation (OO) is a term first coined by one of the Smalltalk inventors, Alan Kay (1969). It was initially conceived as a programming paradigm based on the recognition of real-world communicating objects in computer simulations (Fichman and Kemerer, 1992. Kölling, 1999). OO features were integrated accordingly to platform limitations, in virtually all major programming languages, and toolkits, including the Bio* projects. The Bio* toolkits' usage of OO is commonly hybrid or emulated through modules (Cock *et al.*, 2009; Stajich *et al.*, 2002), mixing objects with primitive data types and hampering the use of reflective functionalities (Maes, 1977). BioSmalltalk benefits from decreased source code verbosity, and its execution in a self-contained snapshot system that promotes run-time adaptability, critical for systems where shutdown cycles cannot be tolerated (Hirschfeld and Lämmel, 2005).

2 FEATURES

2.1 Bioinformatics

BioSmalltalk provides objects to manipulate biological sequences and data from databases like the Entrez system (Schuler *et al.*, 1996). It also contains wrappers for command-line tools like ClustalW (Thompson, 1994) and HMMER (Finn, 2011) sequence visualization and format conversion.

We based implementation on existing FOSS bioinformatics platforms, specifically BioPerl and Biopython, to prevent educational obsolescence, preserving the familiar object model interfaces for experienced bioinformaticians.

BioSmalltalk contains tokenizers, parsers and formatters for common sequence identifiers, FASTA, BLAST and Entrez XML, PHYLIP (Felsenstein, 1989), Arlequin (Excoffier, 2005) and others. Most parsers use PetitParser (Renggli *et al.*, 2010), a dynamically reconfigurable parser library. Additional features can be found in the project documentation. We did a microbenchmark to compare the performance of our library using the script in Figure 1. We have executed the scripts five times immediately after booting without unnecessary processes (Tests were performed on GNU/Linux Debian kernel 2.6.32-358.2.1.el6.x86_64 using an Intel(R) Xeon(R) CPU E5620 at 2.40 GHz, 8 GB DDR3 RAM). Results show that BioSmalltalk has a faster execution time compared with the corresponding BioPerl and Biopython versions. Our approach enabled the removal of unnecessary iterators, thus also reducing the lines of code. Comparison details and scripts are included in the Supplementary Material (Table 1).

2.2 Software engineering

We wrote a cross-platform engine subsystem for enabling multiple interchangeable implementations of packages, which may fail, become unsupported or become too slow. Interchangeable serializers, web client and servers and accessing OS functions were included in our initial release. We have applied design patterns through the library; for example, the *Façade* pattern

*To whom correspondence should be addressed.

```
outFileStream := BioObject newFullFileNamed: 'out.txt'.
msa := BioAlignment new.
(BioNCBIBlastReader newFromXML: 'BS617-Alignment.xml')
 selectedNodes: #('Hit_id' 'Hsp_align-len' 'Hsp_hseq')0
 hitNodesDo: [ : hitNode |
  hitNode hspsDo: [ : hspNode |
    hspNode selectHspAlign = 240 ifTrue: [
     msa addSequence: (BioSequence
           newNamed: hitNode selectAccessionNumber
           sequence: hspNode selectHspHSeq) ]]].
msa asFasta outputTo: outFileStream.
outFileStream close.
```

Fig. 1. A downloaded NCBI BLAST XML result is filtered with alignments matching 240 nucleotide bases. Alignment object is built from sequences and exported as FASTA in a file

Table 1. Microbenchmark results

Environment	LOC	Average execution time (msec)	Peak memory usage (Mbytes)
BioPerl	23	11.496	88.035
Biopython	18	9.595	47.443
BioSmalltalk	11	8.073	45.717

Note: LOC, lines of code.

(Gamma *et al*. 1995) is used to wrap the complex hierarchy of specific parsers. The developer guide provides further information on new engines, testing process and other subsystems.

Maintainability was recognized as an unfilled gap in bioinformatics software development (Umarji and Seaman, 2008). An advantage of BioSmalltalk is relying on a development style that promotes highly factored reusable code (Boehm, 1986) using browsers and inspectors in a targeted navigation manner (Bergel *et al*., 2007; Bunge, 2009), applying automated code refactorings directly through menu options (Opdyke, 1992). This style replaces taking care of boilerplate code, static or primitive-type coercion casting, checking class or function scopes and maintaining directory trees, configuration files or compiler flags.

The software was tested on Windows, Linux and Mac OS X platforms under Squeak and Pharo Smalltalk (Black *et al*. 2009).

3 RESULTS

We delivered an interactive programming system using a fully reflective language for bioinformatics development. We believe that our platform is suitable for a bioinformatics evolution to human-centered long-running software. Of interest for future research is building a user-base and solid automated build process. We are open to collaboration in any of the areas in which BioSmalltalk project can evolve.

Funding: This work is supported by The National Scientific Research Council in Argentina (CONICET), (PIP 2010-2012 N 379).

Conflict of Interest: none declared.

REFERENCES

Bergel,A. *et al*. (2007) Meta-driven browsers. In *Advances in Smalltalk, Proceedings of 14th International Smalltalk Conference*. Springer, pp. 134–156.

Black,A. *et al*. (2009) *Pharo by Example*. Square Bracket Associates, Kehrsatz, Switzerland.

Boehm,B.W. (1986) Understanding and controlling software costs. *10th IFIP World Congress*. Dublin, pp. 703–714.

Bunge,P. (2009) Scripting Browsers with Glamour. PhD Thesis, Master's Thesis, University of Bern, Bern, Switzerland.

Cock,P.J. *et al*. (2009) Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Diemand,A.V. and Scheib,H. (2004) MolTalk: a programming library for protein structures and structure analysis. *BMC Bioinformatics*, **5**, 39.

Excoffier,L.G. *et al*. (2005) Arlequin ver. 3.0: an integrated software package for population genetics data analysis. *Evol. Bioinform. Online*, **1**, 47–50.

Felsenstein,J. (1989) PHYLIP—phylogeny inference package (Version 3.69). *Cladistics*, **5**, 164–166.

Fichman,R. and Kemerer,C. (1992) Object-oriented and conventional analysis and design methodologies. *IEEE Computer*, **25**, 22–29.

Finn,R.D. *et al*. (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res*., **39**, 29–37.

Gamma,E. *et al*. (1995) Design patterns: elements of reusable object-oriented software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goldberg,A. and Robson,D. (1983) *Smalltalk-80: The Language*. Addison-Wesley, Reading, MA.

Hirschfeld,R. and Lämmel,R. (2005) Reflective designs. *IEEE J. Softw*., **152**, 38–51.

Holland,R.C. *et al*. (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–2097.

Kay,A. (1969) *The Reactive Engine*. PhD Thesis, Electrical Engineering and Computer Science University of Utah.

Kölling,M. (1999) The problem of teaching object-oriented programming, Part 1: Languages. *J. Object-Oriented Program*., **11**, 8–15.

Maes,P. (1977) Concepts and experiments in computational reflection. In *OOPSLA '87 Proceedings*, pp. 147–155, ACM, New York, NY, USA.

Opdyke,W.F. (1992) Refactoring Object-Oriented Frameworks. PhD thesis, University of Illinois, IL, USA.

Renggli,L. *et al*. (2010) Practical dynamic grammars for dynamic languages. In *4th Workshop on Dynamic Languages and Applications*. Springer, Málaga, Spain.

Schuler,G.D. *et al*. (1996) Entrez: molecular biology database and retrieval system. *Methods Enzymol*., **266**, 141–162.

Stajich,J.E. *et al*. (2002) The Bioperl toolkit: perl modules for the life sciences. *Genome Res*., **12**, 1611–1618.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting. *Nucleic Acids Res*., **22**, 4673–4680.

Umarji,M. and Seaman,C. (2008) *Informing design of a search tool for bioinformatics*. Retrieved from http://cs.ua.edu/˜SECSE08/Papers/Medha.pdf (30 July 2013, date last accessed).

Wishart,D.S. (1997) A platform-independent graphical user interface for SEQSEE and XALIGN. *Comput. Appl. Biosci*., **13**, 561–562.