BIOINFORMATICS APPLICATIONS NOTE Vol. 28 no. 7 2012, pages 1035–1037
doi:10.1093/bioinformatics/bts080

Sequence analysis

Advance Access publication February 12, 2012

Biogem: an effective tool-based approach for scaling up open source software development in bioinformatics

Raoul J.P. Bonnal[1,*,†], Jan Aerts[2], George Githinji[3], Naohisa Goto[4], Dan MacLean[5], Chase A. Miller[6], Hiroyuki Mishima[7], Massimiliano Pagani[1], Ricardo Ramirez-Gonzalez[8], Geert Smant[9], Francesco Strozzi[10], Rob Syme[11], Rutger Vos[12], Trevor J. Wennblom[13], Ben J. Woodcroft[14], Toshiaki Katayama[15,†] and Pjotr Prins[9,†]

[1]Integrative Biology Program, Istituto Nazionale Genetica Molecolare, Milan 20122, Italy, [2]ESAT/SCD, Faculty of Engineering and IBBT Future Health Department, University of Leuven, Heverlee (Leuven) 3001, Belgium, [3]Parasitology and Immunology Department, KEMRI-Wellcome Trust Research Program, Kilifi, P.O. Box 230 - 80108, Kenya, [4]Department of Genome Informatics, Genome Information Research Center, Research Institute for Microbial Diseases, Osaka University, Suita, Osaka, 565-0871, Japan, [5]The Sainsbury Laboratory, Norwich Research Park, Norwich, NR4 7UH, UK, [6]Biology Department, Boston College, Chestnut Hill, 02467, Massachusetts, USA, [7]Department of Human Genetics, Nagasaki University Graduate School of Biomedical Sciences, Nagasaki, 852-852, Japan, [8]Sequence Informatics, Bioinformatics, The Genome Analysis Centre, Norwich, NR4 7UH, UK, [9]Laboratory of Nematology, Wageningen University, Wageningen, 6708 PB, The Netherlands, [10]CeRSA, Parco Tecnologico Padano, Lodi, 26900, Italy, [11]Department of Environment and Agriculture, Curtin University, Perth, 6102, Western Australia, Australia, [12]NCB Naturalis, Leiden, Postbus 9517, 2300 RA, The Netherlands, [13]Silicon Life Sciences, 55406, Minneapolis, 55406, Minnesota, USA, [14]Department of Biochemistry and Molecular Biology, University of Melbourne, Melbourne, 3010, Australia and [15]Human Genome Center, Institute of Medical Science, University of Tokyo, Tokyo 108-0071, Japan

Associate Editor: Martin Bishop

ABSTRACT

Summary: Biogem provides a software development environment for the Ruby programming language, which encourages community-based software development for bioinformatics while lowering the barrier to entry and encouraging best practices.

Biogem, with its targeted modular and decentralized approach, software generator, tools and tight web integration, is an improved general model for scaling up collaborative open source software development in bioinformatics.

Availability: Biogem and modules are free and are OSS. Biogem runs on all systems that support recent versions of Ruby, including Linux, Mac OS X and Windows. Further information at http://www.biogems.info. A tutorial is available at http://www.biogems.info/howto.html

Contact: bonnal@ingm.org

Received on September 27, 2011; revised on February 8, 2012; accepted on February 9, 2012

1 INTRODUCTION

In biomedical science, new technologies, data formats and methods emerge continuously. Scientists want to take advantage of these developments as soon as possible, which requires bioinformatics software to keep up with new requirements. We support the notion of the Open Bioinformatics Foundation (OBF) that development of collaborative open source software (OSS) is essential for bioinformatics. The OBF represents a number of important projects, such as BioPerl (Stajich *et al.*, 2002), Biopython (Cock *et al.*, 2009), BioRuby (Goto *et al.*, 2010) and BioJava (Holland *et al.*, 2008). These Bio-star (Bio*) projects effectively function as community centres and share a centralized approach in software development with large source code repositories. Bio* projects, generally, aim for consolidated tools, a stable application programming interface (API), and backwards compatibility.

Within the BioRuby project we experienced the drive for stability easily overwhelmed and discouraged developers. Not only because of the complexity of the existing code base, but also because coding standards are enforced, and extensive tests and documentation are required. Furthermore, newly contributed code may be subject to community scrutiny, and in many cases further demands for improving the code follow. The full process introduces a significant delay between initial idea and final acceptance of the code in the main project. Months, even years, may pass between stable releases of main Bio* projects. It may take a long time before a new feature is publicly released.

To scale up collaborative software development in BioRuby, we recognized existing and new developers need to be encouraged to contribute more code. To achieve this, we created Biogem a Ruby application framework for rapid creation of decentralized, internet published software modules written to lower the barrier to entry. Biogem was initially inspired by the R/Bioconductor

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first and last two authors should be regarded as joint First Authors.

© The Author(s) 2012. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from http://bioinformatics.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

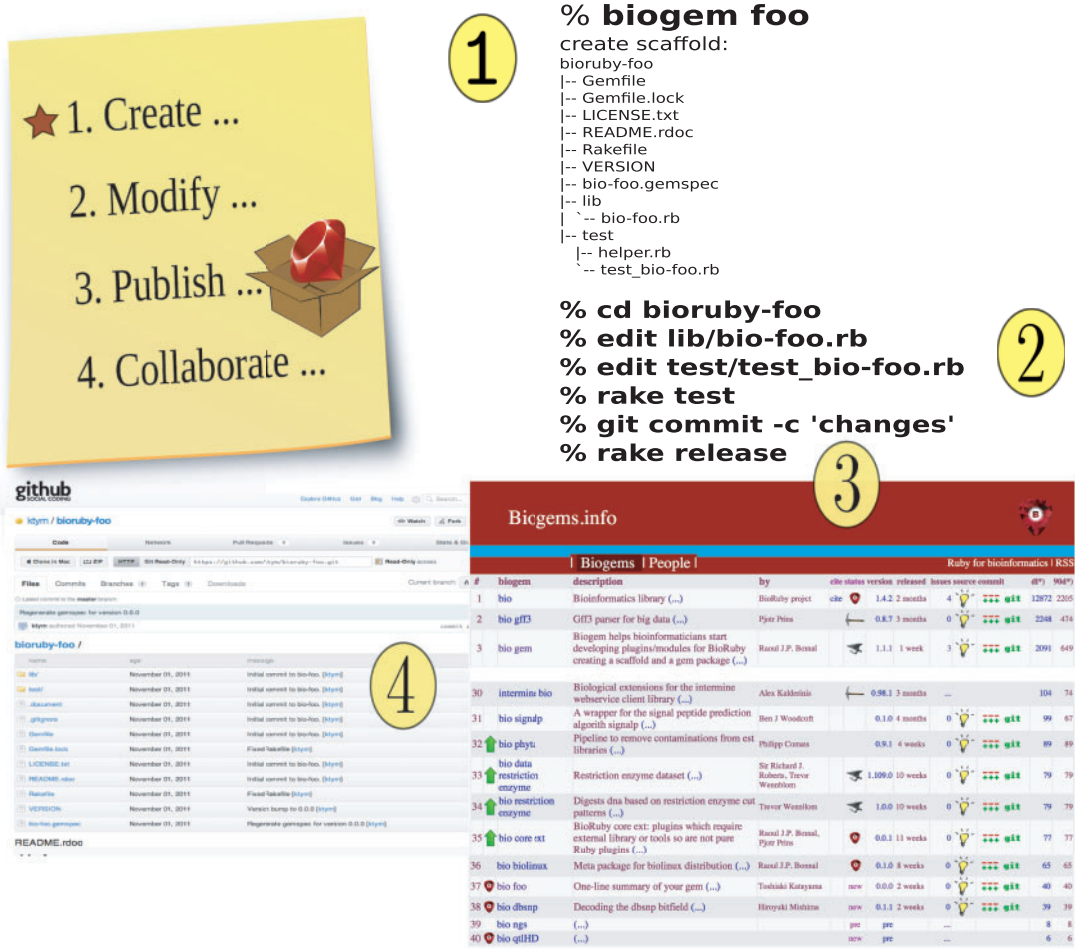

Fig. 1. Biogem eases publication of new bioinformatics Ruby software modules on the Internet, in a few steps. (1) The software generator creates the directory layout and files for a new software module named 'foo'. (2) The developer writes or modifies source code and (3) quickly and easily publishes the source code and module online, for others to read, install and use. Collaboration (4) is facilitated by publishing source code and changes to navigationable websites. Then the workflow continues again at (2). The http://biogems.info website tracks published modules. Popularity of each published module is tracked, as well as source code changes, updates, bugs and issues. Unlike with the practise of publishing scientific papers, collaboration on software often comes *post factum*, i.e. after original publishing of a software module. Therefore, it pays to publish software modules early and often. This is reflected in the Biogem workflow.

packaging system (Gentleman *et al.*, 2004), which encourages software developers to publish software modules independently using simple rules; and Ruby on Rails plugins (Thomas *et al.*, 2006), which provides a software generator and modular software plugin system.

2 FEATURES

For Biogem, we created specific tools to support the creation of bioinformatics software functionalities and to support development 'best practises', i.e. infrastructure for software specification, documentation and tests. We also provide tight web integration based on public websites and services. These websites publish and distribute software modules and give web-based access to source code, complete with revision history (see Fig. 1). Biogem exposes Ruby bioinformatics modules, and makes developer productivity and module popularity visible.

The primary tool of the Biogem framework is a software generator consisting of templates for bioinformatics scripts, source code, software specification, documentation and tests. With the generator, required directories and files are automatically created from templates for a new software module. Templates are included for commonly encountered tasks, such as command line parameter handling, error handling, make files etc.

Another Biogem tool publishes the versioned module with its dependencies on the internet. The published module is immediately available for download and installation to bioinformatics users in the form of a Ruby gem (i.e. an archive of modular Ruby code with all the supporting files and information needed for installation by 'package manager' software). We refer to a Biogem module as a 'BioRuby plugin' if the module extends the BioRuby project. Published software modules are easily repackaged by software distributions, e.g. Debian Bio Med (Möller *et al.*, 2010) and BioLinux (Field *et al.*, 2006).

The Biogem website (see Availability) makes it easy to find and install software modules. The website also allows people to track releases, software dependencies, development activity, outstanding issues, integration test results, documentation and popularity of

Downloaded from http://bioinformatics.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016

Table 1. The introduction of Biogem has led to a broad range of new BioRuby plugins

Name	Description
bio assembly	Read and write assembly data
bio blastxmlparser	Fast, low memory, big data BLAST parser
bio bwa	Burrows Wheeler aligner
bio cnls scraper	Nuclear localisation signal prediction
bio six frame	Sequence translation
bio genomic interval	Detect intervals
bio gff3	Fast, low memory, big data GFF3 parser
bio isoelectric point	Calculate protein isoelectric point
bio kb illumina	Illumina annotations
bio lazyblastxml	Another BLAST XML parser
bio logger	Sane error handling
bio nexml	NeXML support, for phylogenetic data
bio ngs	NGS workflows and display, included support for Bwa, Bowtie, TopHat, and Cufflinks
bio octopus	Transmembrane domain predictor interface
bio restriction enzyme	DNA cutting operations with REBASE
bio samtools	Samtools API
bio signalp	Signal peptide prediction interface
bio sge	Split huge files for cluster computing
bio tm hmm	Transmembrane predictor interface
bio ucsc api	UCSC Genome Database binding

An up-to-date list can be found at http://biogems.info.

published modules. A map shows the location of Biogem developers to help foster a sense of international community.

Biogem encourages software development best practices by providing templates for documentation and multiple test driven development strategies; such as unit tests, behaviour driven development and a natural language parser for software specification (e.g. Chelimsky *et al.*, 2010). A notable difference to the traditional code contribution procedures of the Bio* projects is that best practices are encouraged, rather than enforced.

Templates are also included for certain types of functionality, e.g. to generate portable SQL database handlers, and to build a dynamic website. With Biogem it is possible to create a functional web application, or service, in just a few steps. Generating the different features is handled through work flows (Fig. 1).

We added tutorials for Biogem, which explain the software generators, templates and software publishing. These tutorials are part of the software distribution and available online.

We created 'collections' that bundle important modules together as specific releases. For example, 'bio-core' contains stable modules, and 'bio-core-ext' contains stable modules with bindings to C libraries. Special purpose collections exist such as 'bio-biolinux', which is distributed by the Cloud Biolinux project and merged with the Galaxy CloudMan project (Afgan *et al.*, 2010).

In the first 8 months of the Biogem functionality becoming available, over 20 new modules have been published through Biogem, showing a wide variety of subjects. These modules, for example, target big data handling, next generation sequencing and parsing of bioinformatics data formats (Table 1).

3 CONCLUSION

Biogem provides an environment for rapid bioinformatics software development with a low barrier to entry. Biogem frees potential contributors from code maturity expectations that can be deterring, and encourages Ruby developers to contribute experimental source code early to the BioRuby community. Through Biogem software is published in a modular way, and best practises are encouraged through infrastructure for software specification and testing. All this results in better utilization of existing and new software development manpower, thereby scaling up OSS development in bioinformatics.

We suggest Biogem can serve as a generic model; not by replacing existing Bio* projects, but by supplementing them with a decentralized and evolutionary model for collaborative software development.

ACKNOWLEDGEMENTS

We thank our four reviewers for constructive and detailed comments; reviewers Brad Chapman and Hilmar Lapp identified themselves. We also thank Steffen Möller for comments.

Funding: This work was supported by the Research Council KUL SymBioSys and Flemish Government IBBT (PFV/10/016 to J.A.); the Netherlands Organisation for Scientific Research/TTI Green Genetics (1CC029RP to P.P.); the Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (23791230 to H.M.) and the EC FP7/2007-2013 Marie Curie Fellowship (237046 to R.V.).

Conflict of Interest: none declared.

REFERENCES

Afgan,E. *et al*. (2010) Galaxy CloudMan: delivering cloud compute clusters. *BMC Bioinform.*, (11 Suppl. 12), S4.

Chelimsky,D. *et al*. (2010) *The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends*. Pragmatic Bookshelf Series. The Pragmatic Programmers, LLC.

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

Field,D. *et al*. (2006) Open software for biologists: from famine to feast. *Nat, Biotechnol.*, **24**, 801–803.

Gentleman,R.C. *et al*. (2004) Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biol.*, **5**, R80.

Goto,N. *et al*. (2010) BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**, 2617–2619.

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

Möller,S. *et al*. (2010) Community-driven computational biology with Debian Linux. *BMC Bioinform.*, (11 Suppl. 12), S5.

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

Thomas,D. *et al*. (2006) *Agile Web Development with Rails*. The facets of Ruby series. 2nd edn. Pragmatic Bookshelf.

Downloaded from http://bioinformatics.oxfordjournals.org/ at KU Leuven University Library on October 18, 2016