Bioinformatics Advance Access published August 9, 2012

BioJava: an open-source framework for bioinformatics in 2012

Andreas Prlić [1,*], Andrew Yates [2], Spencer E. Bliven [3], Peter W. Rose [1], Julius Jacobsen [2], Peter V. Troshin [4], Mark Chapman [5], Jianjiong Gao [6], Chuan Hock Koh [7], Sylvain Foisy [8], Richard Holland [9], Gediminas Rimša [10], Michael L. Heuer [11], H. Brandstätter–Müller [12], Philip E. Bourne [13], and Scooter Willis [14]

[1]San Diego Supercomputer Center, University of California San Diego, La Jolla, CA 92093, USA [2]European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton Cambridge CB10 1SD, UK [3]Bioinformatics Program, University of California San Diego, La Jolla, CA 92093, USA [4]College of Life Sciences, University of Dundee, Dundee, DD1 5EH, UK [5]Department of Computer Science, University of Wisconsin-Madison, WI 53706, USA [6]Computational Biology Center, Memorial Sloan-Kettering Cancer Center, New York, NY 10065, USA [7]NUS Graduate School for Integrative Sciences and Engineering, Singapore 117597 [8]Genetics and Genomics Medicine of Inflammation, Montreal Heart Institute, Montreal, Qc H1T 1C8, Canada [9]Eagle Genomics Ltd., Babraham Research Campus, Cambridge CB22 3AT, UK [10]Faculty of Mathematics and Informatics, Vilnius University, LT-03225 Vilnius, Lithuania [11]Harbinger Partners, Inc. St. Paul, MN 55127, USA [12] University of Applied Sciences Upper Austria, 4232 Hagenberg, Austria [13]Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, La Jolla, CA, USA [14]Genomics Core, Scripps Florida, Jupiter, FL 33458, USA

Associate Editor: Dr. Alex Bateman

ABSTRACT

Motivation: BioJava is an open-source project for processing of biological data in the Java programming language. We have recently released a new version (3.0.5), which is a major update to the code base that greatly extends its functionality.

Results: BioJava now consists of several independent modules that provide state of the art tools for protein structure comparison, pairwise and multiple sequence alignments, working with DNA and protein sequences, analysis of amino acid properties, detection of protein modifications, and prediction of disordered regions in proteins, as well as parsers for common file formats using a biologically meaningful data model.

Availability: BioJava is an open-source project distributed under the Lesser GPL (LGPL). BioJava can be downloaded from the BioJava website (http://www.biojava.org). BioJava requires Java 1.6 or higher.

Contact: andreas.prlic@gmail.com All inquiries should be directed to the BioJava mailing lists. Details are available at http://biojava.org/wiki/BioJava:MailingLists

1 INTRODUCTION

BioJava is an established open source project driven by an active developer community (Holland *et al.*, 2008). It provides a framework for processing commonly used biological data and has seen contributions from more than 60 developers in the 12 years since its creation. The supported data range in scope from DNA and protein sequence information up to the level of 3-D protein structures. BioJava provides various file parsers, data models and algorithms to facilitate working with the standard data formats and enables rapid application development and analysis.

The project is hosted by the Open Bioinformatics Foundation (OBF, http://www.open-bio.org), which provides the source code repository, bug tracking database, and email mailing lists. It also supports projects like BioPerl (Stajich *et al.*, 2002), BioPython (Cock *et al.*, 2009), BioRuby (Goto *et al.*, 2010), EMBOSS (Rice *et al.*, 2000) and others.

2 METHODS

Over the last two years, large parts of the original code base have been re-written. BioJava 3 is a clear departure from the version 1 series. It now consists of several independent modules built using Maven (http://maven.apache.org). The original code has been moved into a separate biojava-legacy project, which is still available for backwards compatibility. In the following we describe several of the new modules and highlight some of the new features that are included in the latest version of BioJava.

2.1 Core Module

The core module provides classes to model nucleotide and amino acid sequences and their inherent relationships. Emphasis was placed on using Java classes and method names to describe sequences that would be familiar

*to whom correspondence should be addressed

to the biologist and provide a concrete representation of the steps in going from a gene sequence to a protein sequence to the computer scientist.

BioJava 3 leverages recent innovations in Java. A sequence is defined as a generic interface, allowing the framework to build a collection of utilities which can be applied to any sequence such as multiple ways of storing data. In order to improve the framework's usability to biologists, we also define specific classes for common types of sequences, such as DNA and proteins. One area which highlights this work is the translation engine, which allows the interconversion of DNA, RNA, and amino acid sequences. The engine can handle details such as choosing the codon table, converting start codons to a methionine, trimming stop codons, specifying the reading frame, and handling ambiguous sequences ('R' for purines, for example). Alternatively, the user can manually override defaults for any of these.

The storage of sequences is designed to minimize memory usage for a large collections using a "proxy" storage concept. Various proxy implementations are provided which can store sequences in memory, fetch sequences on demand from a web service such as UniProt, or read sequences from a FASTA file as needed. The latter two approaches save memory by not loading sequence data until it is referenced in the application. This concept can be extended to handle very large genomic data sets, such as NCBI GenBank or a proprietary database.

2.2 Protein Structure Modules

The protein structure modules provide tools for representing and manipulating 3-D biomolecular structures, with the particular focus on protein structure comparison. It contains Java ports of the FATCAT algorithm (Ye and Godzik, 2003) for flexible and rigid body alignment, a version of the standard Combinatorial Extension algorithm (CE) (Shindyalov and Bourne, 1998), as well as a new version of CE that can detect circular permutations in proteins (Bliven and Prlić, 2012). These algorithms are used to provide the RCSB Protein Data Bank (PDB) (Rose *et al.*, 2011) Protein Comparison Tool, as well as systematic comparisons of all proteins in the PDB on a weekly basis (Prlić *et al.*, 2010).

Parsers for PDB and mmCIF file formats (Bernstein *et al.* (1977),Fitzgerald *et al.* (2006)), allow the loading of structure data into a reusable data model. Notably, this feature is used by the SIFTS project to map between UniProt sequences and PDB structures (Velankar *et al.*, 2005). Information from the RCSB PDB can be dynamically fetched without the need to manually download data. For visualization, an interface to the 3-D viewer Jmol (Hanson, 2010) http://www.jmol.org/ is provided. Work is underway for better interaction with the RCSB PDB viewers (Moreland *et al.*, 2005).

2.3 Genome and Sequencing Modules

The genome module is focused on the creation of gene sequence objects from the core module by supporting the parsing of GTF files generated by GeneMark (Besemer and Borodovsky, 2005), GFF2 files generated by GeneID (Blanco and Abril, 2009) and GFF3 files generated by Glimmer (Kelley *et al.*, 2011). The gene sequences can then be written out as a GFF3 format for importing into GMOD (Stein *et al.*, 2002). A separate sequencing module provides memory efficient, low level, and streaming I/O support for several common variants of the FASTQ file format from next generation sequencers. (Cock *et al.*, 2010).

2.4 Alignment Module

The alignment module supplies standard algorithms for sequence alignment and establishes a foundation to perform progressive multiple sequence alignments. For pairwise alignments, an implementation of the Needleman-Wunsch algorithm computes the optimal global alignment (Needleman and Wunsch, 1970), and the SmithWaterman algorithm calculates local alignments (Smith and Waterman, 1981). In addition to these standard pairwise algorithms, the module includes the Guan-Uberbacher algorithm to perform global sequence alignment efficiently using only linear memory (Guan and Uberbacher, 1996). This routine also allows predefined anchors to be manually specified that will be included in the alignment produced. Any

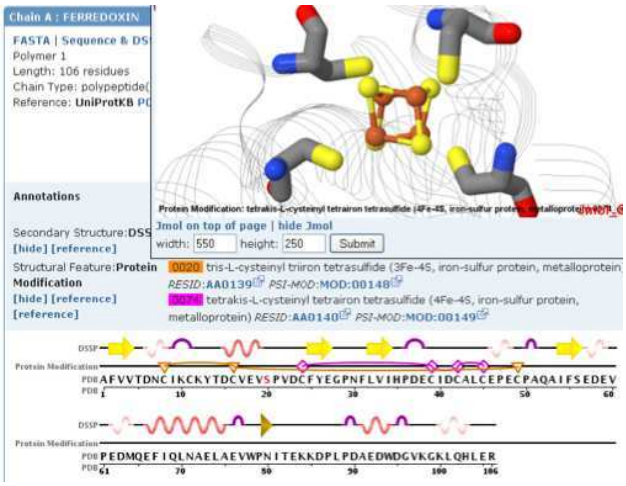

Fig. 1. An example application utilizing the ModFinder module and the Protein Structure module. Protein modifications are mapped onto the sequence and structure of ferredoxin I (PDB ID 1GAO, Chen *et al.* (2002)). Two possible iron-sulfur clusters are shown on the protein sequence (3Fe-4S (F3S): orange triangles/lines; 4Fe-4S (SF4): purple diamonds/lines). The 4Fe-4S cluster is displayed in the Jmol structure window above the sequence display.

of the pairwise routines can also be used to perform progressive multiple sequence alignment. Both pairwise and multiple sequence alignments output to standard alignment formats for further processing or visualization.

2.5 ModFinder Module

The ModFinder module provides new methods to identify and classify protein modifications in protein 3-D structures. More than 400 different types of protein modifications (phosphorylation, glycosylation, disulfide bonds, metal chelation, etc.) were collected and curated based on annotations in PSI-MOD (Montecchi-Palazzi *et al.*, 2008), RESID (Garavelli, 2004), and RCSB PDB (Berman *et al.*, 2000). The module provides an API for detecting protein modifications within protein structures. Figure 1 shows a web-based interface for displaying modifications which was created using the ModFinder module. Future developments are planned to include additional protein modifications by integrating other resources such as UniProt (Farriol-Mathis *et al.*, 2004).

2.6 Amino Acid Properties Module

The goal of the Amino Acid Properties module is to provide a range of accurate physico-chemical properties for proteins. The following peptide properties can currently be calculated: molecular weight, extinction coefficient, instability index, aliphatic index, grand average of hydropathy, isoelectric point, and amino acid composition.

To aid proteomic studies, the module includes precise molecular weights for common isotopically labelled or post-translationally modified amino acids. Additional types of PTMs can be defined using simple XML configuration files. This flexibility is especially valuable in situations where the exact mass of the peptide is important, such as mass spectrometry experiments.

2.7 Protein Disorder module

BioJava now includes a port of the Regional Order Neural Network (RONN) predictor (Yang *et al.*, 2005) for predicting disordered regions of proteins. BioJava's implementation supports multiple threads, making it approximately 3.2-times faster than the original C implementation on a modern quad-core machine.

The Protein Disorder module is distributed both as part of the BioJava library and as a standalone command line executable. The executable is optimized for use in automated analysis pipelines to predict disorder in multiple proteins. It can produce output optimized for either human readers or machine parsing.

2.8 Web Service Access Module

More and more bioinformatics tools are becoming accessible via the Web. As such, BioJava now contains a web services module that allows bioinformatics services to be accessed using REST protocols. Currently, two services are implemented: NCBI Blast via the Blast URLAPI (previously known as QBlast), and the HMMER web service at hmmer.janelia.org (Finn *et al.*, 2011).

3 CONCLUSION

The BioJava 3 library provides a powerful API for analyzing DNA, RNA and proteins. It contains state of the art algorithms to perform various calculations and provides a flexible framework for rapid application development in bioinformatics. The library also provides lightweight interfaces to other projects that specialize in visualization tools. The transition to Maven made managing external dependencies much easier, allowing the use of external libraries without overly complicating the installation procedure for users.

The BioJava project site provides an online cookbook which demonstrates the use of all modules through short recipes of common tasks. We are looking forward to extending the BioJava 3 library with more functionality over the coming years and welcome contributions of novel components by the community.

ACKNOWLEDGEMENTS

We wish to thank everybody who contributed code, documentation or ideas, in particular A. Al-Hossary, R. Thornton, J. Warren, A. Draeger, G. Waldon and G. Barton. Each contribution is appreciated, although the total list of contributors is too long to be reproduced here. Thanks to the Open Bioinformatics Foundation for project hosting.

Funding: AP, PWR and PEB are supported by the RCSB PDB grant NSF DBI 0829586. JG, MC, and CHK have received funding as part of the Google Summer of Code in 2010 and 2011. PT was funded by the Scottish Universities Life Sciences Alliance (SULSA).

REFERENCES

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, **28**, 235–242.

Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, **112**(3), 535–542.

Besemer, J. and Borodovsky, M. (2005). GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Research*, **33**(Web Server issue), W451–W454.

Blanco, E. and Abril, J. F. (2009). Computational gene annotation in new genome assemblies using GeneID. *Methods In Molecular Biology*, **537**(1), 243–261.

Bliven, S. and Prlić, A. (2012). Circular Permutation in Proteins. *PLoS Computational Biology*, **8**(3), e1002445.

Chen, K., Jung, Y.-S., Bonagura, C. A., Tilley, G. J., Prasad, G. S., Sridhar, V., Armstrong, F. A., Stout, C. D., and Burgess, B. K. (2002). Azotobacter vinelandii ferredoxin I: a sequence and structure comparison approach to alteration of [4Fe-4S]2+/+ reduction potential. *The Journal of Biological Chemistry*, **277**(7), 5603–5610.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and De Hoon, M. J. L. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**(11), 1422–1423.

Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., and Rice, P. M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, **38**(6), 1767–1771.

Farriol-Mathis, N., Garavelli, J. S., Boeckmann, B., Duvaud, S., Gasteiger, E., Gateau, A., Veuthey, A.-L., and Bairoch, A. (2004). Annotation of post-translational modifications in the Swiss-Prot knowledge base. *Proteomics*, **4**(6), 1537–1550.

Finn, R. D., Clements, J., and Eddy, S. R. (2011). HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, **39**(Web Server issue), W29–W37.

Fitzgerald, P. M. D., Westbrook, J. D., Bourne, P. E., McMahon, B., Watenpaugh, K. D., Berman, H. M., and Hall, S. R. (2006). Macromolecular dictionary (mmCIF). In S. R. Hall and B. McMahon, editors, *Online*, volume G, pages 295–443. Springer.

Garavelli, J. S. (2004). The RESID Database of Protein Modifications as a resource and annotation tool. *Proteomics*, **4**(6), 1527–1533.

Goto, N., Prins, P., Nakao, M., Bonnal, R., Aerts, J., and Katayama, T. (2010). BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**(20), 2617–2619.

Guan, X. and Uberbacher, E. C. (1996). Alignments of DNA and protein sequences containing frameshift errors. *Computer applications in the biosciences CABIOS*, **12**(1), 31–40.

Hanson, R. M. (2010). Jmol a paradigm shift in crystallographic visualization. *Journal of Applied Crystallography*, **43**(5), 1250–1260.

Holland, R. C. G., Down, T. A., Pocock, M., Prlić, A., Huen, D., James, K., Foisy, S., Dräger, A., Yates, A., Heuer, M., and Schreiber, M. J. (2008). BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**(18), 2096–7.

Kelley, D. R., Liu, B., Delcher, A. L., Pop, M., and Salzberg, S. L. (2011). Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, **40**(1), 1–12.

Montecchi-Palazzi, L., Beavis, R., Binz, P.-A., Chalkley, R. J., Cottrell, J., Creasy, D., Shofstahl, J., Seymour, S. L., and Garavelli, J. S. (2008). The PSI-MOD community standard for representation of protein modification data.

Moreland, J. L., Gramada, A., Buzko, O. V., Zhang, Q., and Bourne, P. E. (2005). The Molecular Biology Toolkit (MBT): a modular platform for developing molecular visualization applications. *BMC Bioinformatics*, **6**(1), 21.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Prlić, A., Bliven, S., Rose, P. W., Bluhm, W. F., Bizon, C., Godzik, A., and Bourne, P. E. (2010). Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics*, **26**(23), 2983–2985.

Rice, P., Longden, I., and Bleasby, A. (2000). EMBOSS: the European Molecular Biology Open Software Suite. *Trends in Genetics*, **16**(6), 276–277.

Rose, P. W., Beran, B., Bi, C., Bluhm, W. F., Dimitropoulos, D., Goodsell, D. S., Prlic, A., Quesada, M., Quinn, G. B., Westbrook, J. D., Young, J., Yukich, B., Zardecki, C., Berman, H. M., and Bourne, P. E. (2011). The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res*, **39**(Database issue), D392—-D401.

Shindyalov, I. N. and Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension {(CE)} of the optimal path. *Protein Eng.*, **11**, 739–747.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C. J., Osborne, B. I., Pocock, M. R., Schattner, P., Senger, M., Stein, L. D., Stupka, E., Wilkinson, M. D., and Birney, E. (2002). The Bioperl toolkit: Perl modules for the life sciences. *Genome Research*, **12**(10), 1611–1618.

Stein, L. D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J. E., Harris, T. W., Arva, A., and Lewis, S. (2002). The Generic Genome Browser: A Building Block for a Model Organism System Database. *Genome Research*, **12**(10), 1599–1610.

Velankar, S., McNeil, P., Mittard-Runte, V., Suarez, A., Barrell, D., Apweiler, R., and Henrick, K. (2005). E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res*, **33**(Database issue), D262—-D265.

Yang, Z. R., Thomson, R., McNeil, P., and Esnouf, R. M. (2005). RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics*, **21**(16), 3369–3376.

Ye, Y. and Godzik, A. (2003). Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19 Suppl 2**, II246—-II255.