Journal of Computational Intelligence in Bioinformatics
ISSN 0973-385X Volume 6, Number 2 (2013) pp. 87-92
© Research India Publications
http://www.ripublication.com/jcib.htm

Role and Challenges of Bioprogramming Languages in the Development of Bioinformatics Tools

Pramod Shinde[1]*, Komal Patil[1], Virochan Chavan[2], Parakh Sharma[1], Jigar Patel[1], Gauri Shende[1], Aparna Patil[1]

[1]Department of Bioinformatics, Guru Nanak Khalsa college, Matunga, Mumbai 400019, India
[2]Department of Bioanalytical Sciences, Guru Nanak Khalsa college, Matunga, Mumbai 400019, India
Corresponding Author: pramodshinde119@gmail.com

ABSTRACT

Biological data is growing in dramatic way. Human genome project and bioinformatics algorithms lead to the development in the computational biology tools and databases. Research in computational biology depends critically on access to biological sequences, databases and outputs of various bioinformatics tools. Various bioprogramming modules are constructed to speed up the bioinformatics tool development. Bioprogramming tools are available for windows, Linux, UNIX and macOS platforms. These modules are easy to use by anyone and can effectively be amalgamated with the tools for the versatile use. These packages still need of continuous development and modification to compete with the expansion of data.

KEYWORDS: bioprogramming, bioinformatics, computational biology, tools

INTRODUCTION

Soft computing with bioprogramming is a consortium of methodologies that work synergistically and provides, in one form or another, flexible information processing capabilities for handling real-life ambiguous situations. The collection of biological data has been increasing at explosive rates due to improvements of existing technologies and the introduction of new ones such as genome sequencing, microarrays, protein sequencing and structure findings(by NMR, X-ray crystallography, modelling, threading etc), sequence characterization etc. processes need the huge amount of computational utilities [1,2]. All the biological data is deposited in the online freely available public repositories. Most biological research involves application of some type of mathematical, statistical, or computational tools to help synthesize recorded data and integrate various types of information in the process of answering a particular biological question [3].

Bioprogramming modules are describing the basic principles of soft computing and demonstrating the various ways in which they can be used for analyzing biological data in an efficient manner [4]. These modules are created in the various bio programming languages. These bioprogramming language modules are constructed by introducing specific packages in the programming languages such as Perl, Python, Ruby and Java etc. Also, BioSQL is developed by enhancing capability to handle the biological data in relational databases. BioLinux is a well featured workstation available for the Linux users as biological data is freely floating in open source environment. As biological data need to handle at different interface level, data transaction uniformality need to maintain and BioXML, BioPax, SBML standards are introduced to tackle uniformality problems.

All these packages are developed in the open source environment so that researchers can contribute and the code is available to anyone interested. BioPerl (www.bioperl.org) is a collection of more than 500 Perl modules for bioinformatics that have been written and maintained by an international group of volunteers and available under Comprehensive Perl Archive Network (CPAN) [5]. BioPython (www.biopython.org/wiki/Main_Page) is a distributed collaborative effort to develop Python libraries and applications which address the needs of current and future work in bioinformatics [6, 7]. The source code is made available under the BioPython License, which is extremely liberal and compatible with almost every license in the world. Similar work is carried to develop BioRuby language for Bioinformatics (www.bioruby.org/rdoc/)[8]. BioPerl, BioPython, BioRuby are the scripting languages in which BioPerl is predominantly preferred by the researchers. These scripts can be amalgamated with other powerful API development languages to develop the potential tools.

BioJava (www.biojava.org/wiki/) was conceived in 1999 by Thomas Down and Matthew Pocock as an Application Programming Interface (API) to simplify bioinformatics software development using Java [9, 10]. The features these languages are enhanced by using Object-oriented programming which is the practice of grouping related tasks together into logical and broadly applicable components [11,12].

The BioSQL (www.biosql.org/wiki/) provides a work platform for storing biological sequences and annotations. BioSQL also provides reusability data for local biological databases and user can very effectively use these relational databases in development of tools [13]. BioSQL provides platforms for BioPython, BioPerl, BioJava and BioRuby languages. The Bio-Linux (www.nebc.nerc.ac.uk/tools/bio-linux/) is a fully featured, powerful, configurable and easy to maintain bioinformatics workstation. It provides more than 500 bioinformatics programs and having various versions separately available for different Linux platforms. It provides a sample data for testing programs, a graphical menu for bioinformatics programs, as well as easy access to the Bio-Linux bioinformatics documentation system. BioLinux is also available for the cloud computing to quickly provision on-demand infrastructures for high-performance bioinformatics computing using cloud platforms [14, 15].

Like XML (eXtensible Markup Language), BioXML (www.bioxml.org/wiki/) is developed for providing a set of standard xml formats for the exchange of biological data [16]. System biology deals with biological macromolecular interaction by biological pathways which are a series of actions among molecules in a cell that leads to a certain product or a change in a cell. There are hundreds of pathway databases are available and to maintain the standard exchange format between these databases SBML (System Biology Markup Language; www.sbml.org/Main_Page) and BioPax (www.biopax.org/) languages are introduced. These pathway standard languages aim to enable integration, exchange, visualization and analysis of biological pathway data [17, 18].

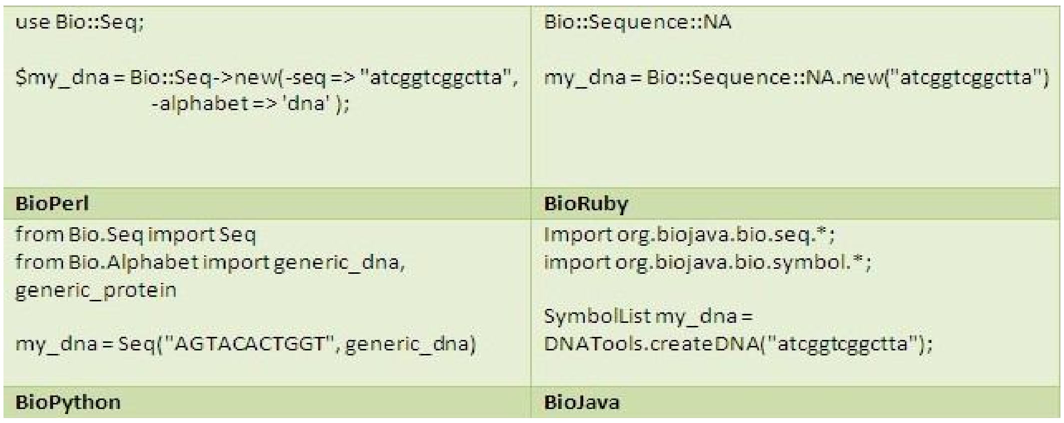
FEATURES

All bioprogramming languages provide essential features like:

- Accessing biological sequence data from local and online databases
- Accessing online sequence alignment tools (like BLAST, FASTA, Clustal etc.) and structure alignment tools (like DALI) as well as to format the results provided by these tools
- Changing formats of database/ file records
- Using web services provided by the various Bio-web portals
- Manipulating and annotating individual sequences
- BioSQL data model are based on the flat files hence, it is very easier for conversion of data and annotations by bioprogramming
- BioSQL can be accessible from PostgreSQL, MySQL, Oracle, HSQLDB, and Apache Derby for the core schema

This programming flexibility provides coders to alter the code available in the packages as per the experimental need. As shown in Table 1, manipulating the biological sequences is very much easy by using Bioprogramming languages.

Table 1: Comparison of coding patterns for defining the DNA sequence in different Bioprogramming languages



use Bio::Seq; $my_dna = Bio::Seq->new(-seq => "atcggtcggctta", -alphabet => 'dna');	Bio::Sequence::NA my_dna = Bio::Sequence::NA.new("atcggtcggctta")
BioPerl	**BioRuby**

from Bio.Seq import Seq
from Bio.Alphabet import generic_dna,
generic_protein
 	
my_dna = Seq("AGTACACTGGT", generic_dna)
	Import org.biojava.bio.seq.*;
	import org.biojava.bio.symbol.*;
	
	SymbolList my_dna =
	DNATools.createDNA("atcggtcggctta");

BioPython	**BioJava**

FUTURE DEVELOPMENT

Because of the special characteristics of biological data, the variety of new problems and the extremely high importance of bioinformatics research, a large number of modules for handling the particular tasks should be continuously developed [19]. Bioprogramming aims to provide scripts and tools that are of use to researchers using bio packages to develop bioinformatics software, regardless of which specialization they may work in [20, 21]. Microarray data analysis is still a complex process and very few bio packages are available to solve gene expression profiles [12, 22]. The reading of outputs from various Blast tools results are still unsupported by Bio packages. Biological system mimicking using computational modeling is constructing multitudes of assumptions and inspiring the biologist to develop new tools [23]. Continuously bioprogramming packages generating is required to tackle with these expansion of multitude tools leading to different data and output formats.

CONCLUSION

Currently, many bioprogramming languages are evolved to solve biological sequences, structure and other data related challenges. These developments lead to increase in the bioinformatics tools and rapid in the processing the computational biology problems by machine learning. . Bioprogramming packages are preferred in tool development using machine learning. In recent years focus is based on Hidden Markov Model, Artificial Neural Network, Server Vector Machines, genetic algorithm etc. to expand the understandings of biological problems.

Biological data is obtained from multiple sources and touches the wide desiplinaries like medicine, disease, bioinformatics etc. Hence, there are shortcomings between the available bioprogramming packages and the variation in the biological data and output formats. Standard formats are introduced in recent years to overcome these limitations. Hence, there is lots of expansion scope in the introduction of new bioprogramming modules.

REFERENCES

[1] Maojo V *et al.* (2006)Designing new methodologies for integrating biomedical information in clinical trials. Methods Inf Med 45: 180–185.

[2] Sayers,E.W. *et al.* (2011) Database resources of the National Center for Biotechnology Information. NAR, 40, D13-25.

[3] Jacob Beal, Andrew Phillips, Douglas Densmore, and Yizhi Cai (2011), High-Level Programming Languages for Biomolecular Systems, in Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology, Springer

[4] Umesh P., Naveen F.(2010), Programming languages for synthetic biology, Syst Synth Biol 4:265–269 DOI 10.1007/s11693-011-9070-y

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

[6] Peter J. A. Cock, Tiago Antao, Biopython: freely available Python tools for computational molecular Biology and bioinformatics, Vol. 25 no. 11 2009, pages 1422–1423 doi:10.1093/bioinformatics/btp163

[7] Chapman,B. and Chang,J. (2000) Biopython: Python tools for computational biology. ACM SIGBIO Newslett., 20, 15–19.

[8] Naohisa Goto, Pjotr Prins (2010), BioRuby: bioinformatics software for the Ruby programming language, Bioinformatics 26 (20): 2617-2619.doi: 10.1093/bioinformatics/btq475

[9] Pocock,M. (2003) Computational analysis of genomes. PhD thesis, University of Cambridge, Cambridge, UK.

[10] Pocock,M. *et al.* (2000) BioJava: open source components for bioinformatics. ACM SIGBIO Newsl., 20, 10–12.

[11] Holland R. C. G., Down T. A. *et al.*(2008), BioJava: an open-source framework for bioinformatics, Vol. 24 no. 18 2008, pages 2096–2097 doi:10.1093/bioinformatics/btn397

[12] Kalpana Raja (2012),Bio-programming prospects of Java: A computational move towards the understanding of the biological aspects of genes and proteins, Indian Journal of Computer Science and Engineering (IJCSE, ISSN : 0976-5166 Vol. 2 No. 6 Dec 2011-Jan 2012

[13] Morgan G. I. Langille1 (2012), MicrobeDB: a locally maintainable database of microbial genomic sequences, Bioinformatics Advance Access published May 9, 2012

[14] Dudley JT, Butte AJ: In silico research in the era of cloud computing. Nat Biotechnol 2010, 28:1181-1185

[15] Konstantinos Krampis, Tim Booth *et al*, Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community, BMC Bioinformatics 2012, 13:42 doi: 10.1186/1471-2105-13-42

[16] Achard F., Vaysseix,G. and Barillot,E. (2001) XML, bioinformatics and data integration. Bioinformatics, 17, 115–125

[17] Hucka M. and Finney A. *et al* (2003), The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models, Vol. 19 no. 4 2003, pages 524–531 DOI: 10.1093/bioinformatics/btg015

[18] Finja Buchel and Clemens Wrzodek (2012), Qualitative translation of relations from BioPAX to SBML qual, Bioinformatics Advance Access published August 24, 2012

[19] Hinsen K. (2000), The molecular modeling toolkit: a new approach to molecular simulations. J. Comp. Chem., 21, 79–85.

[20] Pearson W.R. and Lipman,D.J. *et al.*, (1988) Improved tools for biological sequence analysis. PNAS, 85, 2444–2448.

[21] Rice P. *et al.* (2000) EMBOSS: the European molecular biology open software suite. Trends Genet, 16, 276–277.

[22] Saeed AI, Sharov V, White J *et al* (2003) TM4: a free, open-source system for microarray data management and analysis. Biotechniques, 34:374-378.

[23] Rouchka, E. C. (1999), Pattern Matching Techniques and Their Applications to Computational Molecular, DOI 10.1007/978-3-642-16345-62