Bioinformatics & Knowledge Engineering

Deepak Garg

Thapar Institute of Engineering & Technology, Patiala

Abstract

Two softwares used for sequence alignment and searches are BioJava and BioPerl Comparison between these two tolls is shown. There are several software engineering issues such as software reuse, component-based software engineering, cleanroom engineering. We are discussing here software reuse (reusable objects). What kind and amount of bimolecular data is available and how to encode that data and using which technique (called knowledge engineering), here is one technique to give input of bimolecular data to neural networks on the bases of which to conclude patterns.

1. Introduction

Bioinformatics uses Computer software tools for database creation, data management, data warehousing, data mining. Bioinformatics deals with recording, annotation, storage, analysis, and searching/retrieval of nucleic acid sequence (genes and RNAs), protein sequence and structural information. This includes databases of the sequences and structural information as well methods to access, search, visualize and retrieve the information. It concerns about the creation and maintenance of databases of biological information whereby researchers can both access existing information and submit new entries.The tools developed in the past can be classified as:

Algorithm based: tools belonging to this category embody a deterministic or statistical algorithm, some of which are equipped with visualization and Web Interfaces.

Knowledge based: tools of this category are implemented with background or domain knowledge, and usually borrow techniques developed from the neural networks and machine learning community.

SOFTWARE ENGG ISSUES	REASON
Effective tool development	Biologists use tools to perform data analysis
Internet-based access	Biologists access data and tools via ftp, email
Scalable visualization interface	It allows biological relationships to be modeled in an expressive format.

Table 2

KNOWLEDGE ENGG ISSUES	REASON

The need of machine learning tools	The amount of bimolecular data is enormous and no theory exists for processing the data
Data encoding and knowledge representation.	Good representations are crucial to the success of machine learning.
Feature and knowledge extraction	This allows the automation of Machine learning process.
The need of background knowledge	One should exploit the biological characteristics of the data as much as possible.

2. SE Issues in Bioinformatics

The focus of software engineering research has been shifted from system-oriented tools to user-oriented tools for problem solving. Software reuse has always been a goal in software engineering. Subroutine libraries, for various systems tasks and applications areas, have long been a part of the software development scene. In recent years object-oriented languages and component-based architectures have facilitated development and use of reusable components. Development of reusable components is especially relevant for bioinformatics, where there is heavy use of databases and analysis services over a common domain. Some elements of this domain are *very* heavily used – for example, nucleotide and peptide sequences and alignments. The common tasks in bioinformatics include representing sequences and structures, translating from DNA to amino acid sequences, parsing BLAST reports, reading sequences files in different formats, and converting sequence file formats. Data provided by database services and analysis tools is often used in the context of analysis programs specialized for a particular piece of biological research. There has been a great deal of redundancy in software development – hundreds of programs have modules to represent sequences and alignments, and to read and write them. Reusable components can greatly reduce the amount of redundant effort. Here we discuss some of the current technologies and efforts for reusable software and distributed computing in bioinformatics. The key attributes discussed are the *design* of the software from a *software engineering perspective* and the *appropriateness* from the perspective of molecular biology. Technologies discussed are BioPerl, and BioJava.

2.1 Biojava.org

The BioJava Project is an open-source project dedicated to providing Java tools for processing biological data.
The biojava.org software is based on the Java programming language. Java is modern, object-oriented with good support and a broad user community. The language has features for large-scale software engineering – it separates interface definition from implementation, which encourages modularity and allows alternative implementations. It has support for packaging and distribution of components in the *package* construct and *java archive* files. Java programs are portable between systems through the Java virtual machine.
Biojava.org has developed interfaces for sequences and features, including representation, manipulation, and IO for external file formats and databases, including Blast, Meme.

2.2 Bioperl.org

The BioPerl project is a coordinated effort to collect computational methods routinely used in bioinformatics into a set of standard CPAN-style, well documented, and freely available Perl modules.
Perl is a scripting language with strong support for text processing and directory/file manipulation. It has support for objects. Data structures like lists and associative arrays (Perl hashes) are fairly easy to use, but have idiosyncratic syntax and semantics. Perl modules are extremely easy to install, and Perl modules for a wide variety of applications are readily available on the CPAN archive.
Bioperl.org has implemented sequence objects, sequence alignment objects, and a BLAST wrapper object. Sequence objects have operations for translation, reverse complement, subsequence extraction. They support extended alphabets, and there is strong support for reading and writing different file formats and for database access. There are two kinds of sequence objects – a basic object, and a heavyweight object with support for sequence features and annotations. There is a BLAST wrapper object, with operations for reading a BLAST report and parsing it for its contents, and for writing BLAST reports and formatting them as HTML. There is some support to represent motifs as regular expressions. In addition, there are design considerations for wrapper objects for other sequence analysis utilities based on the BLAST wrapper, and for protein structure.

Table 3

Comparison	**BioPerl**	**BioJava**
Data import facility	Comes with SeqIO module to import sequences. Supported formats are:Fasta,SwissProt, Genbank, SCF, GCG, raw.	Comes with IO package to import sequences. Supported formats are:Fasta,EMBL, Genbank
Data export facility	Same as import	None available
Support for types, number and size of sequences	Protein/DNA/RNA Large number of seqs. Supported. Very large sequences probably supported via Ensemble extensions.	Protein/DNA/RNA Heavyweight list of symbols (a symbol is a residue). Large number and size of sequences excluded probably hard to achieve with good efficiency.
Support for analysis tools	Blast, HMMER, patterns	Meme, others

3. KE Issues in Bioinformatics

Many of the knowledge-based bioinformatics tools are built using neural networks and machine learning techniques. One important issue, in applying neural networks to biosequences analysis, is how to encode the biosequences, i.e., how to represent the biosequences as the input of the neural networks. Good input representations make it easier for the neural networks to recognize the underlying regularities. Thus, good input representations are crucial to the success of the

neural network learning. One of the encoding methods is orthogonal encoding. In orthogonal encoding, nucleotides or amino acids in a biosequences are viewed as unordered categorical values, and are represented by C dimensional orthogonal binary vectors, where C is the cardinality of the 4-letter DNA alphabet D = {A, T, G, C} or the cardinality of the 20-letter amino acid alphabet A = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. That is, we use C binary (0/1) variables, among which only one binary variable is set to 1 to represent one of the C possible categorical values and the rest are all set to 0. For instance, we represent the nucleotide
A by "1000", and amino acid Y by "00000000000000000001". The orthogonal encoding was frequently used in the early 1990s. The orthogonal encoding requires that the bio-sequences be equal in length, or one must sample the bio-sequences of variable lengths by a window of fixed size. Disadvantage is that it wastes a lot of input units in the input layer of a neural network. For instance, for a protein sequence of 100 amino acids, 2000 input units are required to represent the protein sequence. This requires many neural network weight parameters as well as many training data, making it difficult to train the neural network.
An alternative encoding method is to use high-level features extracted from biosequences. The high-level features should be relevant and biologically meaningful. By "relevant", we mean that there should be high mutual information between the features and the output of the neural network, where the mutual information measures the average reduction in uncertainty about the output of the neural network given the values of the features. By "biologically meaningful", we mean that the features should represent the biological characteristics of the sequences.

4. Conclusion

Several SE and KE issues in bioinformatics have been addressed. New research directions such as gene expression, genome warehousing, protein synthesis, RNA processing and structure prediction, are emerging. These areas have recently gained significant attention from both computer and natural scientists. As the bioinformatics era is starting, we anticipate that SE and KE technologies are becoming increasingly important in developing complex, intelligent, and large-scale software for biological information processing. Bioperl and BioJava uses the software principle software reuse**.** New research is being implemented such as how can we improve the efficiency of these tools by comparing their features.

References

[1] Java and BioJava: BioJava home page java.sun.com Java from the source – Sun Microsystems.

[2] J. D. Hirst and M. J. E. Sternberg. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. Biochemistry, 31:7211- 7218, 1992.

[3] J. T. L. Wang, S. Rozen, B. A. Shapiro, D. Shasha, Z. Wang, and M. Yin. New techniques for DNA sequence classi_cation. Journal of Computational Biology, 6(2):209-218, 1999.

[4] Q. Ma, J. T. L. Wang, and C. H. Wu. Application of Bayesian neural networks to biological data mining: A case study in DNA sequence classification. In Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, pages 23-30, 2000.