Handbook of Software Engineering and Knowledge Engineering
Vol. 0, No. 0 (2000) 000–000

SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING ISSUES IN BIOINFORMATICS

JASON T. L. WANG, QICHENG MA, KATHERINE G. HERBERT
Department of Computer and Information Science
New Jersey Institute of Technology
University Heights, Newark, New Jersey 07102, USA

In this chapter we address several SE and KE issues in bioinformatics. We review and compare two most widely used bioinformatics tools for sequence alignment and searches. Then, we address the need of background knowledge for processing biomolecular data. Next, we discuss the design and status of a bioinformatics infrastructure, called Genome Mining, developed in our lab. Finally, we conclude the chapter by pointing out some future research directions.

Keywords: Software development; machine learning; neural networks; biomolecular data processing.

1. Introduction

Bioinformatics, or computational biology, refers to an emerging, interdisciplinary field in which computer technology, including software, hardware and algorithms, is applied to solving problems arising in biology. One subject, of particular interest in the field, is to develop tools for processing biomolecular data [3, 18, 19, 20]. These data include DNA (deoxyribonucleic acid), RNA (ribonucleic acid), protein sequences, and their two-dimensional (2D) and three-dimensional (3D) structures.

DNA has a twisted double helical structure. Each strand of the DNA double helix is a polymer built from four components, called *nucleotides*: `A`, `T`, `C`, and `G` (the abbreviations for adenine, thymine, cytosine, and guanine). The two strands of DNA are complementary: whenever there is a `T` on one strand, there is an `A` in the corresponding position on the other strand; whenever there is a `G` on one strand, there is a `C` in the corresponding position on the other. DNA can be represented by a sequence of these four letters, or *bases*.

Like DNA, RNA is a long molecule but is usually single stranded, except when it folds back on itself. It differs chemically from DNA by containing ribose sugar instead of deoxyribose and containing the base uracil (`U`) instead of thymine. Thus, the four bases in RNA are `A`, `C`, `G`, and `U`.

A protein is also a polymer, constructed by hundreds or thousands of amino acids. The most popular representation model for biologists to describe a protein is to use the sequence. A protein sequence is made up of 20 amino acids, each represented by a letter: alanine (`A`), cysteine (`C`), aspartic acid (`D`), glutamic acid (`E`), phenylalanine (`F`), glycine (`G`), histidine (`H`), isoleucine (`I`), lysine (`K`), leucine

(L), methionine (M), asparagine (N), proline (P), glutamine (Q), arginine (R), serine (S), threonine (T), valine (V), tryptophan (W), and tyrosine (Y).

As a result of the Human Genome Project and other initiatives, biomolecular data accumulate at an accelerating rate. For example, the Protein Information Resource (PIR) database [2], maintained at the National Biomedical Research Foundation of Georgetown University Medical Center and accessible at `http://pir.georgetown.edu/`, now contains 190,392 sequences (release 65, as of September 1, 2000).*It is therefore essential to have effective tools for processing these data. Data processing in this context includes classifying and aligning sequences, detecting similarities, finding protein coding regions in DNA sequences, and predicting molecular structure and function. Computational tools designed to improve these processes contribute to our understanding of life as well as to the discovery of drugs.

In examining the tools developed in the past, we can roughly classify them into two categories:

- 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.

While the techniques underlying these tools are of interest, our goal here is to address some software engineering (SE) and knowledge engineering (KE) issues in bioinformatics; we refer the reader to other introductory texts [1, 21] for algorithmic details of the tools. Table 1 (Table 2, respectively) summarizes the SE (KE, respectively) issues and explains why they arise. In addressing the SE issues, we will survey and compare two most prominent tools for sequence alignment and searches (Section 2). In addressing the KE issues, we point out the need of background knowledge and review a tool built using neural networks and machine learning techniques (Section 3). As described in these two sections, the most effective computational framework is based on incorporating and combining different tools together, as the tools often complement each other. Section 4 then presents the design of Genome Mining, an infrastructure built in our lab for biomolecular data processing. Finally, Section 5 concludes the chapter and points out some future research directions.

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 [17]. Software tools in bioinformatics, for example, are targeted toward solving problems arising in biology. Among them, BLAST and FASTA are two most eminent tools, both being freely accessible on the

*In release 40 of 1995, the database has 24,569 sequences. In release 29 of 1992, the database has 8,309 sequences.

SE Issues	Reason
Effective tool development	Biologists use tools to perform data analysis.
Internet-based access	Biologists access data and tools via ftp, email, and World Wide Web.
Scalable visualization interface	It allows for biological relationships to be modeled in an expressive format.

Table 1: Software Engineering Issues in Bioinformatics

KE Issues	Reason
The need of machine learning tools	The amount of biomolecular 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 for automating the machine learning process.
The need of background knowledge	One should exploit the biological characteristics of the data as much as possible.

Table 2: Knowledge Engineering Issues in Bioinformatics

Internet. Essentially, the two tools are "sequence alignment programs." Using local alignment algorithms [1, 21], they try to find the best alignment between a query sequence and every sequence in a database. Using various heuristics, they try to keep the search time within a reasonable time frame. While BLAST and FASTA perform essentially the same tasks, each tool has particular searches that it is better suited to perform [8].

2.1. *BLAST Tool*

BLAST, or Basic Local Alignment Search Tool, was first developed in 1990 using Altschul's method to search for similarities between a query and all the sequences in a database. It is a set of five similarity search programs designed to explore all of the available sequences in a database regardless of whether the query is protein or nucleic acid. BLAST's algorithm essentially looks for matches by first looking for small segments of the input sequence to match with other sequences. It then builds from those matched regions to the largest ungapped regions it can find, cf. Figure 1. Often, a score is assigned to a matching sequence that rates its possibility as a match. These scores are then interpreted statistically. This statistical interpretation makes it easier to determine what is a valid match and what is not a valid match.

The most recent version of BLAST can be accessed at the Web site of the National Center for Biotechnology Information (NCBI). The URL is `http://www.ncbi`

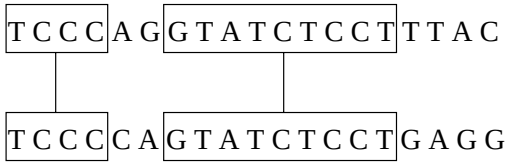

Figure 1: An alignment between two sequences produced by BLAST. Matched regions are highlighted and connected by an alignment line.

.nlm.nih.gov/BLAST/. At this Web site, a scientist has a choice of four different methods for accessing BLAST. The most popular and easiest way to access BLAST is through the World Wide Web interface. At the BLAST Web site, there is an easy-to-use Web page that guides the scientist through the search. Through drop-down boxes, the scientist can choose which program he or she would like to use and then which database he or she would like to search. Then the scientist can enter a protein or nucleic acid sequence in a text box. If the scientist is unfamiliar with the BLAST search programs, BLAST will revert to a default setting for the search. For more precise searches, one can customize a search. Besides the World Wide Web interface, there is also a Network BLAST. This program allows the scientist to run remote searches from his or her computer with his or her computer as a client. However, when using this option of BLAST, one should be aware that some security problems can arise, especially if one is using confidential sequences.

Another way to use BLAST is to download a stand-alone version. This version allows the scientist to implement searches on private databases. Finally, for the scientist who does not have Internet access, there is an EMAIL BLAST. Essentially, the scientist sends a specially formatted email to NCBI with the query sequence. The scientist then receives an email back from NCBI with the results of the search.

2.2. *FASTA Tool*

FASTA is a set of programs initially developed in 1988 using Pearson and Lipman's method to search for similarities between one sequence and any group of sequences of the same type [15]. FASTA searches through databases by initially choosing very small portions of the input sequence to match exactly against sequences in the database. It then begins to work out from those exact matches to find larger ungapped alignments. Finally, FASTA joins these alignments into gapped alignments and calculates a possible score and statistical representation of that score. The output usually is first represented in a histogram followed by an analysis of the search results.

There are many executions of FASTA located on the World Wide Web. The primary site for FASTA is William Pearson's Web site at the University of Virginia. The URL is `http://www.med.virginia.edu/medicine/basic-sci/biochem/faculty/pearson.html`. At this site, the scientist can access a Web-based interface for FASTA. It provides over 20 databases for search as well as an option to search

```
>carAB-P1, promoter
AAAAAAATCCCGCCATTAAGTTGACTTTTAGCGCCCATATCTCCAGAATG
GCCGCCGTTTGCCA
```

(a)

```
>JC4383 3 ' -phosphoadenosine-5 '-phosphosulfate synthetase - spoonworm (Urchis caupo)
MAFLPNGQLATNVTFQTQHVSRAKRGQVLGQRGGFRGCTVWFTGLSGAGKTTISFALE
EYLVSQGIPTYSLDGDNVRHGLNKNLGFTQEDREENIRRISEVAKLFADGGIVCLTSFISP
```

(b)

```
JC4383
```

(c)

Figure 2: Different formats for inputting a query sequence; (a) inputting a DNA sequence; (b) inputting a protein sequence; (c) inputting a sequence using its ID in an existing protein database.

specific proteomes and genomes. Furthermore, the scientist can use default search settings as well as customize his or her searches. Also, there is a version of FASTA one can download for one's own use [15].

Besides the FASTA Web site at the University of Virginia, other groups have developed their own FASTA interface. The interface at the European Bioinformatics Institute's Web site, `http://www.ebi.ac.uk/fasta3/`, is one such example. The interface is highly interactive and allows the user to just use default settings or to specify many variables as well as offering about 30 different databases for search. Also, the scientist can specify whether he or she wants the results emailed to him or her or would like them interactively displayed at the Web site. Moreover, it allows the scientist to choose whatever format he or she would like to input the query sequence in (see Figure 2 for some different input formats). Once the scientist begins the query, the site maintains a clock as to how long the query has been running and instructions to email the query if the search takes too long. The output is displayed in a traditional FASTA output format.

There are other Web sites that maintain versions of FASTA. Most of these sites can be found by doing a simple search from any Web search engine. Most of the differences in these Web sites are just how user friendly the interfaces are. When looking for a FASTA tool to use, it would help to first check Pearson's Web site for the most recent version of the tool—this would help to choose the best possible tool for a search. Also, Bioinformatica, located at `http://www.bioinformatica.com/`, provides many tools and resources for anyone doing research in the bioinformatics field.

2.3. *Comparison of BLAST and FASTA*

While BLAST and FASTA essentially perform the same tasks, each tool performs better in certain queries than in others. The decision is ultimately determined by what kind of search the scientist is looking to do, time constraints, and what database the scientist wishes to search. One of the biggest differences between BLAST and FASTA is time. BLAST was specifically designed to be a speedy search without sacrificing sensitivity. Generally, BLAST is much faster than FASTA. BLAST searches usually take a few minutes while FASTA searches can take significantly longer. Therefore, if time is an important concern while doing a search, then BLAST is the better tool to use.

Another big difference between the two tools is the database access each provides. BLAST can use the non-redundant databases provided by NCBI. For a similar search, FASTA would have to access multiple databases. Moreover, BLAST offers various search modes that allow the scientist more flexibility in his or her search. FASTA doesn't offer this flexibility in some searches and it may require another computer program to translate an input into a form FASTA can read. This makes FASTA very unwieldy to use in certain searches and can add a lot of time and overhead to a search. Also, when using default settings, BLAST tends to be more sensitive than FASTA in protein searches. BLAST's algorithm allows for gapped matches. This allows BLAST to include sequences that, while not being a perfect match, may be a significant match for the query. FASTA's algorithm requires a perfect match during the first stage of the search that can force the tool to overlook some significant matches.

However, when working with nucleic acids, FASTA tends to be more sensitive than BLAST. Since FASTA allows for smaller word searches in the first phase of the search, it is more inclined to find more precise matches. However, this type of search does take time and BLAST can be adequately sensitive for most nucleic acid searches. In the latest version of BLAST, the developers included the ability to search only portions of the databases. This feature has been available for a while on FASTA. If one plans to do a partial database search (such as searching GenBank for mammals), one may want to keep in mind that this type of search is new for the BLAST tool and use the FASTA tool as well to get comprehensive search results.

Table 3 summarizes the comparison of BLAST and FASTA. In general, most users implement a BLAST search using default settings as a first step in a database search. After viewing the results from this search, he or she then decides on whether the information is reliable enough for what they need or if they need to obtain more sensitive data through customized BLAST searches or default or customized FASTA searches.

It is worth pointing out that BLAST/FASTA searches can help to do data mining in biomolecular sequences. For example, a typical data mining problem is to perform classifications. One approach for protein sequence classification is to compare an unlabeled sequence S with the sequences in the target class and the sequences in the non-target class using BLAST or FASTA. One then assigns S to the class containing the sequence best matching S. A similar approach could also

	BLAST	FASTA
speed (using GenBank)	a few minutes	several hours
protein similarity searches	more sensitive	less sensitive
DNA similarity searches	less sensitive	more sensitive
database access	can access more non-redundant databases through NCBI	must access multiple databases to perform searches
search modes	provides facilities to make searches more flexible	may use another program to translate input
partial database searches	feature recently included in tool	feature available for a long period of time

Table 3: Comparison of BLAST and FASTA

be used for DNA sequence classification and recognition [20].

2.4. *Summary*

In reviewing the two most widely used bioinformatics tools, BLAST and FASTA, we conclude that a successful tool must be user-friendly, flexible, usable, and accessible. This tool should be widely available through ftp, email, and the World Wide Web. There is often a tradeoff between time and precision. For example, FASTA is more time consuming but yields more precise results than BLAST. Performance is often data dependent. For example, BLAST is better for protein searches while FASTA is better for nucleic acids sequences. In many cases, tools complement each other. Therefore, the best result can be achieved by combining multiple tools in different ways as suggested by Gaeta [8].

One shortcoming of many of the existing alignment tools is the lack of a visualization interface. Visualization [4, 5], or icon-based navigation and browsing, has been an active research area in software development and engineering. In bioinformatics, only a few areas, such as sequence classification, exploit visualization technology. The most common way to visualize the biomolecular data is to use dendrograms (rooted trees) [9]. Essentially, in this method, the comparisons are performed, the information clustered, and then organized meaningfully into the dendrogram structure. The key advantage to the dendrogram structure is that it allows for biological relationships to be modeled in an expressive format.

While this approach provides methods for representing biological information meaningfully, there are also drawbacks. If the comparison relation is symmetric, then isomorphisms occur. Moreover, the structure cannot be scaled up for large amounts of data. Therefore, different visualization techniques are being investigated to avoid such problems. Current methods being explored involve using three-dimensional space to visualize distances between sequences directly. These methods essentially design distance measures, take these measures and map the sequences to points in 3D space, locate points accordingly, and then visualize these points. As

far as the alignment tools are concerned, some sites of FASTA have implemented a rudimentary interface for displaying histograms and search results. However, how to apply advanced visualization technology to building more user-friendly bioinformatics tools with classification and clustering dendrograms remains to be an open research problem.

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 biosequence 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 [10].

One of the encoding methods is *orthogonal encoding* [14]. In orthogonal encoding, nucleotides or amino acids in a biosequence 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 $\mathcal{D} = \{\texttt{A, T, G, C}\}$, or the cardinality of the 20-letter amino acid alphabet $\mathcal{A} = \{\texttt{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$. 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 [6, 11]. Figure 3 shows an example of the orthogonal encoding of a DNA sequence.

The orthogonal encoding requires that the biosequences be equal in length, or one must sample the biosequences of variable lengths by a window of fixed size. Another 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 reflect the biological characteristics of the sequences. These characteristics are often referred to as the background knowledge in the development of bioinformatics tools.

3.1. *Background Knowledge*

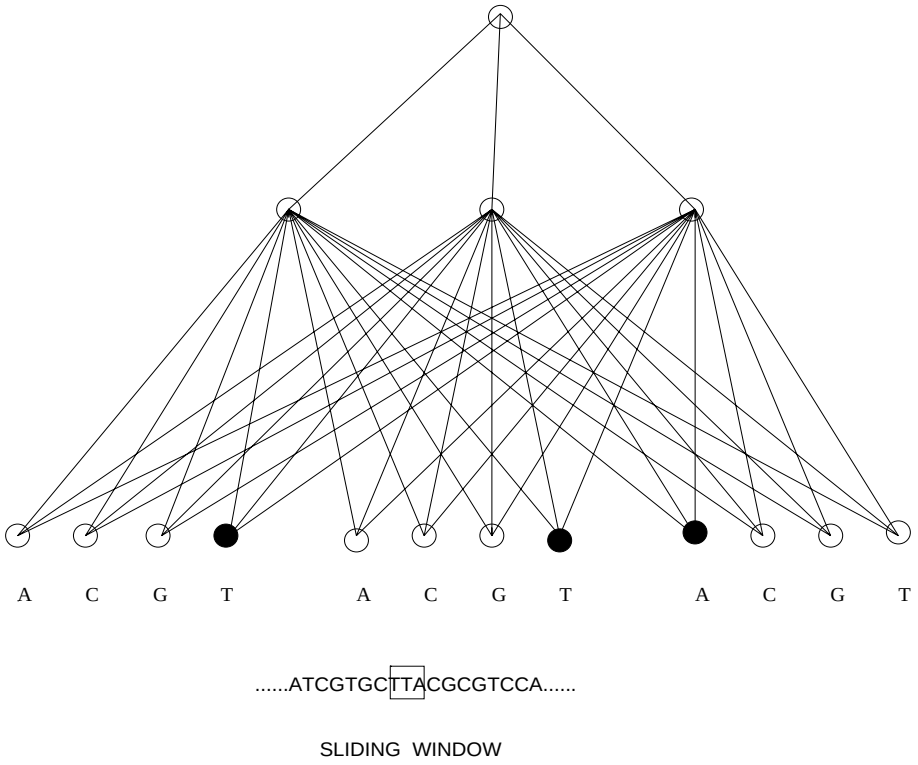

Figure 3: An example of the orthogonal encoding of a DNA sequence.

We use E. Coli promoter recognition to illustrate what is meant by background knowledge. The E. Coli promoter is located immediately before the E. Coli gene. Thus, successfully locating the E. Coli promoter conduces to identifying the E. Coli gene. The uncertain characteristics of the E. Coli promoters contribute to the difficulty in the promoter recognition. The E. Coli promoters contain two binding sites to which the E. Coli RNA polymerase, a kind of protein, binds [12]. The two binding sites are the -35 hexamer box and the -10 hexamer box, respectively. Each binding site consists of 6 bases (nucleotides). The central nucleotides of the two binding sites are roughly 35 bases and 10 bases, respectively, upstream of the transcriptional start site. The transcriptional start site is the first nucleotide of a codon where the transcription begins; it serves as a reference point (position +1). The consensus sequences, i.e., the prototype sequences composed of the most frequently occurring nucleotide at each position, for the -35 binding site and the -10 binding site are `TTGACA` and `TATAAT`, respectively. But none of the promoters can exactly match the two consensus sequences. The average conservation is about 8 nucleotides, meaning that a promoter sequence can match, on average, 8 out of the 12 nucleotides in the two consensus sequences. Figure 4 shows an example promoter sequence with the -35 binding site being `TAGCGA` and the -10 binding site

Figure 4: An example promoter sequence. Regions are highlighted by upper case letters. The -54 region, -44 region, -35 box, -29 region, -22 region, -10 box, and +1 region are `CTTTGTAGC`, `CTTTCAC`, `TAGCGA`, `AACG`, `GAATGG`, `AAAGAT` and `CA`, respectively. Particularly important regions (binding sites) are discussed in the text.

being `AAAGAT`. The conservation here includes only 6 nucleotides.

The two binding sites are separated by a spacer. The length of the spacer has an effect on the relative orientation between the -35 region and the -10 region. A spacer of 17 nucleotides is most probable. The promoter sequence in Figure 4 has a spacer of 17 nucleotides. Another spacer between the -10 hexamer box and the transcriptional start site also has a variable length. The most probable length of this spacer is 7 nucleotides. The promoter sequence in Figure 4 has a spacer of 6 nucleotides. In general, the distance between the -10 binding site and the transcriptional start site varies from 3 to 11 bases. The distance between the -35 binding site and the -10 binding site varies from 15 to 21 bases. These varying distances render promoter recognition difficult, as both the contents and positions of the binding sites are uncertain.

Many promoter sequences have the pyrimidine (`C` or `T`) at the position -1 (one nucleotide upstream of the transcriptional start site), while the purine (`A` or `G`) is at the transcriptional start site (position +1). The +1 region includes the nucleotides at the position -1 and the transcriptional start site. The promoter sequence in Figure 4 has a nucleotide `C` at the position -1 and a nucleotide `A` at the transcriptional start site. To develop bioinformatics tools for recognizing promoters, one has to exploit the characteristics of the E. Coli promoters.

3.2. *Knowledge Extraction and Representation*

Knowledge extraction and representation is an important process in developing knowledge-based bioinformatics tools. We use our recently developed tool for E. Coli promoter recognition [13] to illustrate this process. Our knowledge of the nucleotide probability distributions in the two binding sites of promoter sequences is represented in the Position Weight Matrix (PWM) [16]. We use a machine learning technique, more precisely the expectation-maximization (EM) algorithm [7], to extract the knowledge of the nucleotide probability distributions in the two binding sites. Based on the knowledge, we are able to precisely locate the binding sites of the promoter sequences. We also develop feature extraction algorithms to represent a DNA sequence as a vector of high-level features extracted from the sequence. The feature values are then fed into a neural network (NN).

By considering different combinations of features, we develop three basic programs for promoter recognition. We found experimentally that a combination of the three basic programs outperforms each individual program. This result is consistent with the observation in Section 2.4, where we pointed out that bioinformatics tools often complement each other, and a combination of the tools usually achieves the best result.

4. The Genome Mining Project

The purpose of the Genome Mining project, accessible at `http://www.cis.njit.edu/~eservice`, is to apply data mining and knowledge engineering techniques to genome data processing. The project is to develop a Web-based server, with an advanced visualization interface, that allows the user to perform common genome mining activities, including sequence classification and clustering as well as pattern discovery. The user can run the programs provided by the Genome Mining toolbox remotely on the Web or can download these programs to a local site. The toolbox is connected to major genome data processing centers around the world.

Currently the Genome Mining toolbox has two components. The first component is the NN promoter recognition tool described in Section 3.2. The second component is a protein sequence classification tool, which takes as the input a protein sequence, extracts features from the protein sequence, and feeds these feature values to a trained neural network. The neural network can classify the input protein sequence and tell if it belongs to one of the globin, kinase, ras or ribitol superfamilies in the PIR protein database [2] at the National Biomedical Research Foundation of Georgetown University Medical Center.

Figure 5 illustrates the design of the Genome Mining toolbox. In the figure, a user submits a query sequence to the toolbox through the World Wide Web. Our Web server accepts the sequence, which is processed by the protein classification tool or the promoter recognition tool. Both tools have access to the underlying database containing training sequences and users' profiles. After processing, the result is sent to the Web server, which then returns the result to the user.

In addition to the promoter recognition tool and protein sequence classification tool, we are developing new tools for gene detection and for phylogenetic analysis in plants. These tools will be integrated into the Genome Mining toolbox in the near future.

5. Conclusion

In this chapter we have addressed several SE and KE issues in bioinformatics. Besides topics mentioned in the chapter, new research directions, such as DNA chips, DNA probe arrays, 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

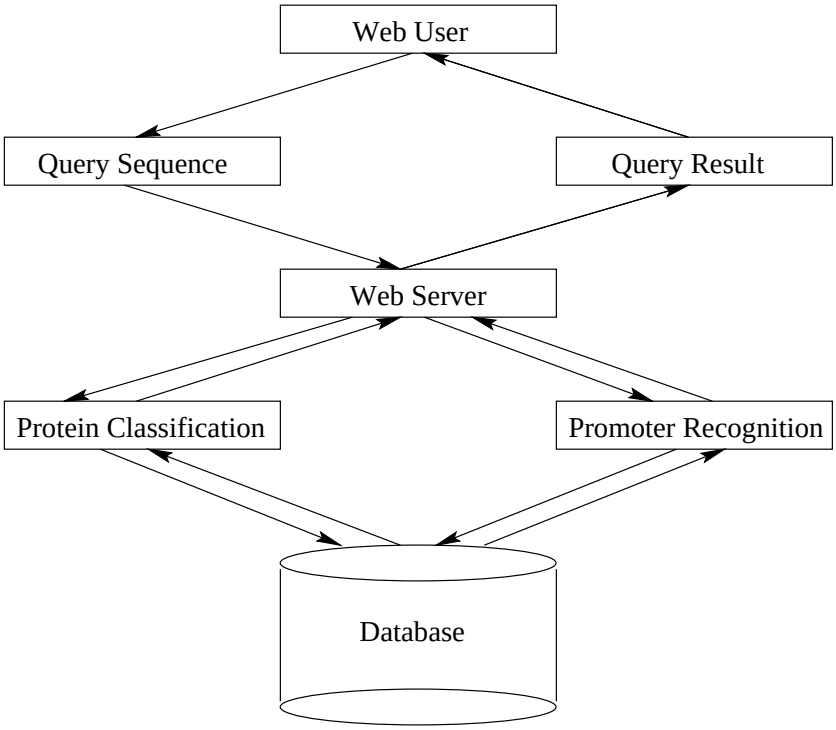

Figure 5: The design of the Genome Mining toolbox.

increasingly important in developing complex, intelligent, and large-scale software for biological information processing.

Acknowledgments

We thank the anonymous reviewers for their thoughtful comments, which helped to improve the presentation and quality of the paper.

1. P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. The MIT Press, Cambridge, Massachusetts, 1998.
2. W. C. Barker, J. S. Garavelli, D. H. Haft, L. T. Hunt, C. R. Marzec, B. C. Orcutt, G. Y. Srinivasarao, L. S. L. Yeh, R. S. Ledley, H. W. Mewes, F. Pfeiffer, and A. Tsugita. The PIR-international protein sequence database. *Nucleic Acids Research*, 26(1):27–32, 1998.
3. C.-Y. Chang, J. T. L. Wang, and R. K. Chang. Scientific data mining: A case study. *International Journal of Software Engineering and Knowledge Engineering*, 8(1):77–96, 1998.
4. S. K. Chang (ed.). *Principles of Visual Programming Systems*. Prentice-Hall, 1990.
5. S. K. Chang (ed.). *Visual Languages and Visual Programming*. Plenum Publishing Corporation, 1990.
6. M. W. Craven and J. W. Shavlik. Machine learning approaches to gene recognition. *IEEE Expert*, 9(2):2–10, 1994.

7. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39:1–38, 1977.
8. B. A. Gaeta. Database similarity searching using BLAST and FASTA. *Australasian Biotechnology*, 1995.
9. D. Gilbert, M. Schroeder, and J. van Helden. Interactive visualization and exploration of biological data. In *Proceedings of the 2nd International Workshop on Biomolecular Informatics*, 2000.
10. H. Hirsh and M. Noordewier. Using background knowledge to improve inductive learning of DNA sequences. In *Proceedings of the 10th Conference on Artificial Intelligence for Applications*, pages 351–357, 1994.
11. 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.
12. S. Lisser and H. Margalit. Compilation of E. Coli mRNA promoter sequences. *Nucleic Acids Research*, 21(7):1507–1516, 1993.
13. 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.
14. D. W. Opitz and J. W. Shavlik. Connectionist theory refinement: Genetically searching the space of network topologies. *Journal of Artificial Intelligence Research*, 6:177–209, 1997.
15. W. R. Pearson. FASTA programs at the University of Virginia, 1997. URL: `http://alpha10.bioch.virginia.edu/fasta/`.
16. R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12(1):505–519, 1984.
17. J. E. Urban and P. O. Bobbie. Software productivity: Through undergraduate software engineering education and CASE tools. In *The Impact of CASE Technology on Software Processes,* (ed. D. Cooke), pages 327–347, World Scientific Publishing Co., Inc., 1994.
18. J. T. L. Wang, T. G. Marr, D. Shasha, B. A. Shapiro, and G.-W. Chirn. Discovering active motifs in sets of related protein sequences and using them for classification. *Nucleic Acids Research*, 22(14):2769–2775, 1994.
19. J. T. L. Wang, T. G. Marr, D. Shasha, B. A. Shapiro, G.-W. Chirn, and T. Y. Lee. Complementary classification approaches for protein sequences. *Protein Engineering*, 9(5):381–386, 1996.
20. J. T. L. Wang, S. Rozen, B. A. Shapiro, D. Shasha, Z. Wang, and M. Yin. New techniques for DNA sequence classification. *Journal of Computational Biology*, 6(2):209–218, 1999.
21. J. T. L. Wang, B. A. Shapiro, and D. Shasha (eds.). *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications*. Oxford University Press, New York, New York, 1999.