Role of Sequence Data Mining in Bioinformatics

Ms. Salma Mohd. Shafi
Head, Department of Computer Science
Bhilai Mahila Mahavidyalaya, Bhilai-490009
Sheikhsalma10@gmail.com

Abstract

Sequence data, and the ability to utilize this hidden knowledge, creates a significant impact on many aspects of our society. Examples of sequence data include DNA, protein, customer purchase history, web surfing history, and more. Sequence Data Mining provides balanced coverage of the existing results on sequence data mining, as well as pattern types and associated pattern mining methods. Sequence Data Mining is designed for professionals working in bioinformatics, genomics, web services, and financial data analysis. The task of sequential data mining is a data mining task specialized for analyzing sequential data, to discover sequential patterns.

Keyword: *Sequence data, DNA, Bioinformatics, Genomics.*

1. Introduction

Recent progress in data mining research has led to the development of numerous efficient and scalable methods for mining interesting patterns in large databases. In the mean time, recent progress in biology, medical science, and DNA technology has led to the accumulation of tremendous amounts of bio-medical data that demands for in-depth analysis. The question becomes how to bridge the two fields, data mining and bioinformatics, for successful mining of bio-medical data. In this abstract, we analyze how data mining may help bio-medical data analysis and outline some research problems that may motivate the further developments of data mining tools for bio-data analysis.

I will now explain the task of sequential data mining with an example. Consider the following sequence database, representing the purchases made by customers in a retail store.

SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

This database contains four sequences. Each sequence represents the items purchased by a customer at different times. A sequence is an ordered list of item sets (sets of items bought together). For example, in this database, the first sequence (SID 1) indicates that a customer bought some items *a* and *b* together, then purchased an item *c*, then purchased items *f* and *g* together, then purchased an item *g*, and then finally purchased an item *e*.

sequential data mining is being used to find subsequences that appear often in a sequence database, that are common to several sequences. Those subsequences are called the frequent sequential patterns. For example, in the context of our example, sequential

pattern mining can be used to find the sequences of items frequently bought by customers. This can be useful to understand the behavior of customers to take marketing decisions.

To do sequential pattern mining, a user must provide a sequence database and specify a parameter called the minimum support threshold. This parameter indicates a minimum number of sequences in which a pattern must appear to be considered frequent, and be shown to the user. For example, if a user sets the minimum support threshold to 2 sequences, the task of sequential pattern mining consists of finding all subsequences appearing in at least 2 sequences of the input database. In the example database, 29 subsequences met this requirement. These sequential patterns are shown in the table below, where the number of sequences containing each pattern (called the *support*) is indicated in the right column of the table.

Pattern	Sup.
$\langle\{a\}\rangle$	3
$\langle\{a\},\{g\}\rangle$	2
$\langle\{a\},\{g\},\{e\}\rangle$	2
$\langle\{a\},\{f\}\rangle$	3
$\langle\{a\},\{f\},\{e\}\rangle$	2
$\langle\{a\},\{c\}\rangle$	2
$\langle\{a\},\{c\},\{f\}\rangle$	2
$\langle\{a\},\{c\},\{e\}\rangle$	2
$\langle\{a\},\{b\}\rangle$	2
$\langle\{a\},\{b\},\{f\}\rangle$	2
$\langle\{a\},\{b\},\{e\}\rangle$	2
$\langle\{a\},\{e\}\rangle$	3
$\langle\{a,b\}\rangle$	2
$\langle\{b\}\rangle$	4
$\langle\{b\},\{g\}\rangle$	3
$\langle\{b\},\{g\},\{e\}\rangle$	2
$\langle\{b\},\{f\}\rangle$	4
$\langle\{b\},\{f,g\}\rangle$	3
$\langle\{b\},\{f\},\{e\}\rangle$	2
$\langle\{b\},\{e\}\rangle$	3
$\langle\{c\}\rangle$	2
$\langle\{c\},\{f\}\rangle$	2
$\langle\{c\},\{e\}\rangle$	2
$\langle\{e\}\rangle$	3
$\langle\{f\}\rangle$	4
$\langle\{f,g\}\rangle$	3
$\langle\{f\},\{e\}\rangle$	2
$\langle\{g\}\rangle$	3
$\langle\{g\},\{e\}\rangle$	2

For example, the patterns <{a}> and <{a}, {g}> are frequent and have a support of 3 and 2 sequences, respectively. In other words, these patterns appears in 3 and 2 sequences of the input database, respectively. The pattern <{a}> appears in the sequences 1, 2 and 3, while the pattern <{a}, {g}> appears in sequences 1 and 3. These patterns are interesting as they represent some behavior common to several customers. Of course, this is a toy example. Sequential pattern mining can actually be applied on database containing hundreds of thousands of sequences.

2. Methods

Alignment of Biological Sequences

The problem of alignment of biological sequences can be described as follows: Given two or more input biological sequences, identify similar sequences with long conserved subsequences. If the number of sequences to be aligned is exactly two, it is called pair wise sequence alignment; otherwise, it is multiple sequence alignment. The sequences to be compared and aligned can be either nucleotides (DNA/RNA) or amino acids (proteins). For nucleotides, two symbols align if they are identical. However, for amino acids, two symbols align if they are identical, or if one can be derived from the other by substitutions that are likely to occur in nature. There are two kinds of alignments: local alignments versus global alignments. The former means that only portions of the sequences are aligned, whereas the latter requires alignment over the entire length of the sequences. For either nucleotides or amino acids, insertions, deletions, and substitutions occur in nature with different probabilities. Substitution matrices are used to represent the probabilities of substitutions of nucleotides or amino acids and probabilities of insertions and deletions. Usually, use the gap character, "□", to indicate positions where it is preferable not to align two symbols. To evaluate the quality of alignments, a scoring mechanism is typically defined, which usually counts identical or similar symbols as positive scores and gaps as negative ones. The algebraic sum of the scores is taken as the alignment measure. The goal of alignment is to achieve the maximal score among all the possible alignments

The BLAST Local Alignment Algorithm

The BLAST algorithm was first developed by Altschul, Gish, Miller, et al. around 1990 at the National Center for Biotechnology Information (NCBI). The software, its tutorials, and a wealth of other information can be accessed at *www.ncbi.nlm.nih.gov/BLAST/*. BLAST finds regions of local similarity between bio sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as to help identify members of gene families. The NCBI website contains many common BLAST databases. According to their content, they are grouped into nucleotide and protein databases. NCBI also provides specialized BLAST databases such as the vector screening database, a variety of genome databases for different organisms, and trace databases. BLAST applies a heuristic method to find the highest local alignments between a query sequence and a database. BLAST improves the overall speed of search by breaking the sequences to be compared into sequences of fragments (referred to as words) and initially seeking matches between these words. In BLAST, the words are considered as *k*-tuples. For DNA nucleotides, a word typically consists of 11 bases (nucleotides), whereas for proteins, a word typically consists of 3 amino acids. BLAST first creates a hash table of neighborhood (i.e., closely matching) words, while the threshold for "closeness" is set based on statistics. It starts from exact matches to neighborhood words. Because good alignments should contain many close matches, and use statistics to determine which matches are significant. By hashing, I find matches in $O(n)$ (linear) time. By extending matches in both directions, the method finds high-quality alignments consisting of many high-scoring and maximum segment pairs. There are many versions and extensions of the BLAST algorithms. For example, MEGABLAST, Discontiguous MEGABLAST, and BLASTN all can be used to identify a nucleotide sequence. MEGABLAST is specifically designed to efficiently find long alignments between very similar sequences, and thus is the best tool to use to find the identical match to a query sequence. Discontiguous MEGABLAST is better at finding nucleotide sequences that are similar, but not identical, to a nucleotide query. One of the important parameters governing the sensitivity of BLAST searches is the length of the initial words, or *word size*. The word size is adjustable in BLASTN and can be reduced from the default value to a minimum of 7 to increase search sensitivity. Thus BLASTN is better than MEGABLAST at finding alignments to related nucleotide sequences from other organisms. For protein searches, BLASTP, PSI-BLAST, and PHI-BLAST are popular. Standard protein-protein BLAST (BLASTP) is used for both identifying a query amino acid sequence and for finding similar sequences in protein databases. Position-Specific Iterated (PSI)-BLAST is designed for more sensitive protein-protein similarity searches. It is useful for finding very distantly related proteins. Pattern-Hit Initiated (PHI)-BLAST can do a restricted protein pattern search. It is designed to search for proteins that contain a pattern specified by the

user and are similar to the query sequence in the vicinity of the pattern. This dual requirement is intended to reduce the number of database hits that contain the pattern, but are likely to have no true homology to the query.

Program	Query sequence	Target database
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Nucleotide in 6 reading frame	Protein
TBLASTN	Protein	Nucleotide in 6 reading frame
TBLASTX	Nucleotide in 6 reading frame	Nucleotide in 6 reading frame

Figure 1 Search possibilities in the BLAST program

BLAST uses a heuristic algorithm that makes it possible to search a huge database in a very short period of time by using a query sequence. The high speed of the algorithm stems from the fact that the query sequence is divided into short „words" that are used, instead of the full-length sequence, during the alignment process. These words are searched in the database first (called „seeding", i.e. finding the best local alignments). The most relevant hits are then scored with the help of a scoring matrix, extended to neighboring words, and finally assembled and compiled into a final list of similarity hits. It is important that the query sequences must be in the so-called FASTA. The FASTA format is shown in Figure 2.

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
        (the first line can be omitted)
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
IENY
```

Figure 2. The FASTA sequence format

3. Result

Bioinformatics analysis of protein sequences

The wide range of *in silico* analysis possibilities of protein sequences is summarised in Figure 3. Many of these analyses can be performed also with nucleic acid sequences. Sequences can be compared to each other and to full databases. The physical and structural/functional properties of polypeptide chains can be predicted via this analysis. Sequence comparisons (alignments) were described in the previous section (BLAST and ClustalW programs). During the so-called profile analysis, the analysed sequences are compared to secondary databases that contain information about protein structural families, structural and functional domains, modules, phosphorylation, glycosylation and other posttranslational modification consensus sequences. Many online programs are available on the internet that can search secondary databases. For instance, the InterProScan profile analysis program can be used to search the InterPro secondary database (in fact it is a "superdatabase" of several individual derived databases)

maintained by the EBI. Another example is the PhosSitePlus database that can be searched by any query sequence to predict phosphorylation or other posttranslational modification sites.

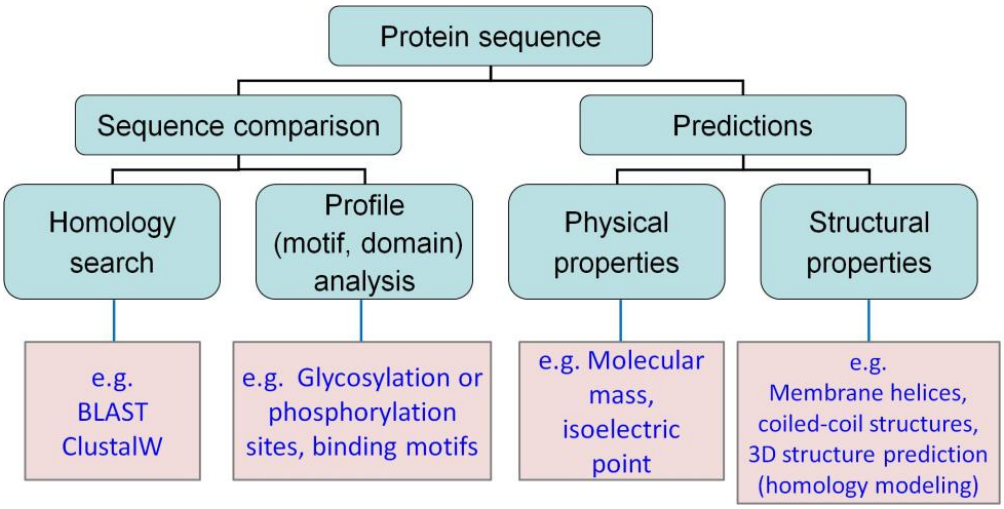

Figure 3. The wide range of *in silico* analysis possibilities of protein sequences. (Most of these options are also available for nucleic acid sequences.)

4. Conclusions

Both data mining and bioinformatics are fast expanding research frontiers. It is important to examine what are the important research issues in bioinformatics and develop new data mining methods for scalable and effective bio-data analysis. I believe that the active interactions and collaborations between these two fields have just started and a lot of exciting results will appear in the near future.

5. References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD'00*, pp. 439–450, Dallas, TX, May 2000.

[2] A. Baxevanis and B. F. F. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins (2nd ed.)*. John Wiley & Sons, 2001.

[3] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or howto build a data quality browser. In *SIGMOD'02*, pp. 240–251, Madison, WI, June 2002.

[4] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probability Models of Proteins and Nucleric Acids*. Cambridge University Press, 1998.

[5] W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction*. Springer-Verlag, New York, 2001.

[6] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, NewY ork, 2001.

[8] A. M. Lesk. *Introduction to Bioinformatics*. Oxford University Press, 2002.

[9] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB'01*, pp. 381– 390, Rome, Italy, Sept. 2001.

[10] H. Wang, J. Yang, W. Wang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *SIGMOD'02*, pp. 418–427, Madison, WI, June 2002.

[11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2001.

[12] J. Yang, P. S. Yu, W. Wang, and J. Han. Mining long sequential patterns in a noisy environment. In *SIGMOD' 02*, pp. 406–417, Madison, WI, June 2002.

[13] Hipp, Jochen , Guntzer, Ullrich and Nakhaeizadeh, Gholamreza, "Algorithms for Association Rule Mining – A general Survey and Comparison". *SIGKDD explorations*, Vol 2, Issue – 1, pp 58 – 63, Mar – 2004.

[14] Mount DW. 2000. *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY. Chapter 1.