International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.1,No.1, 2011

DOI: 10.7321/jscse.v1.n1.3

A Data Mining Approach To Gen Dynamic Behavioral Process

*Mehdi Alimi Motlagh Fard[1], Hamid Reza Ranjbar[1], Abbas Davani[1], Mehdi Sadegh Zadeh[2]

[1]Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Booshehr, Iran

[2]Department of Engineering, Islamic Azad University, Iran

Alimi@LianPro.com

Abstract. Biology systems theory and cell biology have enjoyed a long relationship, which, in the context of systems biology, has received renewed interest in recent years by computer science as bioinformatics models. It has often been noted that computer simulation, by providing explicit hypotheses for a particular system and bridging across different levels of organization, can provide an organizational focus which can be leveraged to form substantive hypotheses. Simulations lend meaning to data and can be updated and adapted as further data comes in. Systems biology is concerned with the dynamic behavior of biochemical reaction networks within cells and in cell populations. The biologist's conceptual frameworks, in which to identify the variables of a biochemical reaction network and to describe their relationships, are pathway maps. A principal goal of systems biology is therefore to turn these static maps into dynamical models. In this paper introduces a data mining approach can be use to process of dynamic behavior of a biochemical network from different perspectives. Most bioinformatics tools require specialized input formats for sequence comparison and analysis. This is particularly true for molecular phylogeny programs, which accept only certain formats. In addition, it is often necessary to eliminate highly similar sequences among the input, especially when the dataset is large. Moreover, most programs have restrictions upon the sequence name.

Keywords: Bioinformatics, Data Mining, Dynamic Behavioral

1. Introduction

Molecular phylogenetic is a based fundamental approach studying species evolution and gene functions and has many models. Many phylogenetic analysis programs are available, but each program often requires a particular type of input sequence format. Most programs have restrictions regarding the allowable length of sequence name and characters used. Automatically trimming the sequence names may lose important information, such as species names. Most often the trimmed names are redundant, which cannot be accepted by any analysis program. In addition, some programs do not accept sequence names with special characters. Furthermore, the input may contain identical or highly similar sequences which need to be removed to save computational resources and improve resolution of the resulting trees. Then we need tools to discovery of behavioral of molecular phylogenetic. In this paper we introduce a data mining approach.

Knowledge discovery and data mining is a process of seeking patterns among the masses of data that can be stored and organized in modern computer infrastructure. Knowledge discovery and data-mining arose in the commercial sector, where information was amassed for accounting and legal reasons over decades before being recognized as a valuable resource, figuratively a gold-mine of information. In the scientific realm, similar techniques have been developed and adapted over the past decade as massive information veins associated with the genomes of human, fly, mouse, and others have come on-line. As suggested by its commercial history, data mining grew out of database: data had been collected and now something was to be done with it. This history resulted in data mining techniques arising apart from the underlying database approach. In this paradigm the database provides data for data mining but is not itself altered by the data-mining endeavor as shown in figure 1. This limitation has been recently addressed by proposing the concept of inductive databases, which utilizes a two-way flow of information between database and data mining tools [1]. The inductive database will include metadata calculated from the base data, which is then used as base data for further exploration. Scientific data mining differs in several ways from the more highly developed commercial applications [2]. Differences include a higher reliance on complex numerical manipulations and less clear-cut goals, requiring that data be analyzed and reanalyzed from different perspectives. In these respects, the scientific knowledge discovery and data mining process is more free form than the commercial variety. Neurobiologists are faced with the intellectual aim of understanding nervous systems, perhaps as complex a task as science faces. At the same time, modern techniques, including the genome projects, are making more and more information available, raising the question of what can best be done with it. In addition to the sociological perils common to cooperation in any scientific field, the emerging field of neuron informatics must also confront unusual interoperability challenges due to the diverse types of data generated. These arise from different techniques, as well as from the many orders of magnitude in time and space covered by different investigations of even a single subsystem [3].

Data mining tools are typically chosen on an ad-hoc basis according to the task. These tools include various algorithmic constructions as well as traditional statistical techniques. Although some statistical procedures are used, the data-mining enterprise differs from statistics in that the data are multi-valued and do not generally fit a recognizable statistical distribution [4]. Thus statistical procedures that depend on the distribution, such as the Student's test, often cannot be used. Those that are used are non-parametric. For example, mean and standard deviations lose meaning when working with a bimodal distribution.

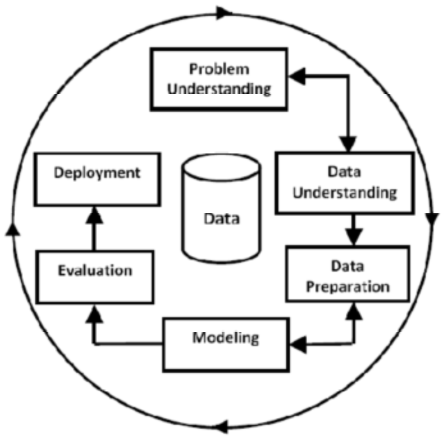

Figure 1. Data Mining Process

Many data mining tools in the commercial world are text-oriented or symbolic, providing clustering or classifications by meaning or symbols.[5] This symbol-oriented approach extends into science in the realm of genomics and proteomics, whose objects of interest are described by a circumscribed list of symbols representing nucleotides and amino-acids[6,7]. In neuroscience, spatially oriented tools, some developed from geography, can be utilized in areas such as development [8]. Neuroanatomy [9,10,11], and imaging.[12,13] Taxonomic thinking, involving anthologies and part-whole relations, require semantically-oriented tools.[14] Other areas of neuroscience, such as electrophysiology, are primarily numerical. Numerical data mining tools include numerical integration and differentiation [15, 16], wavelet and spectroscopic analyses [17], numerical classification methods such as principal component and independent component analysis [18] and standard statistical methods. Spike trains, being a time-series of discrete events, require yet other approaches for analysis [19].

Life is strongly associated with organization and structure [20].With the completion of 1000 genomes project, the project is estimated to generate about 8.2 billion bases per day, with the total sequence to exceed 6 trillion nucleotide bases. The DNA molecule is made up of a concatenation of four different kinds of nucleotides namely: Adenine, Thymine, cytosine and Guanine (A, T, C, and G). General purpose compression algorithms do not perform well with biological sequences. Giancarlo et al. [21] have provided a review of compression algorithms designed for biological sequences.

Finding the characteristics and comparing Genomes is a major task [22, 23]. In mathematical point of view, compression implies understanding and comprehension [24]. Compression is a great tool for Genome comparison and for studying various properties of Genomes. DNA sequences, which encode life, should be compressible. It is well known that DNA sequences in higher eukaryotes contain many tandem repeats, and essentials genes have many copies. It is also proved that genes duplicate themselves sometimes for evolutionary purposes. All these facts conclude that DNA sequences should be compressible. The compression of DNA sequences is not an easy task [25, 26, 27]. DNA sequences consists of only four nucleotides bases {a,c,g,t}. Two bits are enough to store each base. The standard compression software's such as "compress", "gzip", "bzip2", "WinZip" expanded the DNA genome file more than compressing it. Most of the Existing software tools worked well for English text compression [28] but not for DNA Genomes. Increasing genome sequence data of organisms lead DNA database size two or three times bigger annually. Thus it becomes very hard to download and maintain data in a local system. Other algorithms specifically designed for DNA sequences compression did not manage to achieve average compression rate below 1.7 bits/base. Algorithms for Compressing DNA sequences, such as Ziv-Lempel compression algorithms [29, 30]. Bio compress, Gen compress and DNA compress compress DNA sequences.

2. Method

The standard method for detecting exact duplicates is to sort the dataset and then to perform an exact matching with the neighbor records to determine whether two records are duplicates or not. Different authors have extended this approach for detecting the similar/approximate duplicate records in databases. These approaches vary by the amount of domain knowledge supplied by the user. Some of these approaches are domain-specific and/or assumes that the user for each application domain will supply the domain knowledge. In [31] proposed a framework for data cleaning. They addressed the problem of transitivity rule in the field of similar-duplicate detection, and proposed a method for avoiding this problem. Still, their proposed method is domain dependent and requires some rules and certainty factor (*c.f.*). To specify these rules, domain knowledge is necessary. It also requires user involvement for defining the *c.f.* Wang *et al*. proposed another rule based duplicate detection methodology [32].

They used experts' knowledge to create rules and key from a set of attributes when the global key is absent.

Some authors also proposed to create production rules based on domain-specific knowledge. However, creating rules is often time consuming. Certainly, duplicate detection systems should use as less domain knowledge as possible [33]. In addition, the result must be acceptable enough for keeping the time duration in a tolerable range.

The reason is that usually data mining processes are applied on databases, which contain huge amount of data.

In [34] proposed some of the earlier works for similar-duplicate detection. All these works are almost similar and based on Merge/Purge technique. However, their proposed Merge/Purge technique gives solution only for domain-specific problems. The authors used a window-based system to limit the number of comparisons required for a record with its neighbor records. It utilized equational theory that needs to be changed for each domain and also requires domain knowledge. In addition, it used a sorting technique based on keys selected by user. This technique fails, if the user makes a poor choice to sort the data. To increase the efficiency of the data cleaning method, Lee *et al.* proposed a token-based technique for data preprocessing. This technique requires external source files, which contain useful information to remove typical errors from the dataset. Unfortunately, it can be used only for some particular domain-specific problems, and for most cases, it is not possible to get such an external source file. For sorting the dataset, they proposed a method similar to the method proposed in [34]. Their sorting method involved three stages. In the first stage, the data in each field of the selected attribute are tokenized, next the tokens are internally sorted and then they are used for sorting the dataset. This method fails, if the record contains errors in the selected attribute. In [35] proposed a completely domain-independent work. Their proposed sorting method treats each record as one long string and sorts them lexicographically reading from left to right in one pass and right to left in another pass. The authors also proposed a priority queue based technique for limiting the number of comparisons. To group similar records they used pure transitivity rule, but they also mentioned some problems of using transitivity rule for detecting similar-duplicates. In [36], the authors presented a concept in which prediction is made by assigning weight proportional to the importance of information and using co-occurrence and textual similarity functions. However, this work is partially domain-independent and the memory and time requirement for performing the required operations is very high. Hylton proposed an algorithm for identifying and merging related bibliographic records, which are in different formats. It was designed to work using authors' names and titles. It works only for bibliographic records; hence, it is domain-specific. Winkler proposed a method, which can be applied for a limited range of very good quality lists that have known matching characteristics. Weis *et al.* proposed an industry scale duplicate detection system, which is an extension of the domain-independent approach proposed for detecting duplicates in XML data. However, the proposed industry-scale duplicate detection system is totally domain-dependent and requires rules and the knowledge of domain experts'. Similar-string matching is one of the key requirements for similar-duplicate detection systems. Several works are available on similar/approximate string matching in the literature. Unfortunately, most of them are designed to solve particular problems. The well-known Boyer and Moore's algorithm [37] is very effective to find a pattern in a string, but it is a kind of exact matching algorithm and does not hold for similar-string matching.

Du *et al.* proposed an approximate string matching algorithm to solve only dictionary problems. Smith- Waterman algorithm is another well-known algorithm, which was originally developed to compare DNA or protein sequences. The most popular method for similar-string matching is based on edit-distance, which is the minimum number of unit operations (for example, add, delete, insert) needed to perform on individual characters to transform one string to the other. These types of functions are also called classical edit-distance functions, or textual similarity functions. Monge showed that recursive algorithm gives more accurate result for approximate string matching than previous classical algorithms. Still, the complexity of the proposed algorithm is very high due to its recursive comparisons of sub-strings. On the other hand, Udechukwu *et al.* claimed that their proposed Positional algorithm improves the result of the recursive algorithm. Conversely, experimental result shows that, for some cases, the original Positional algorithm fails to return an acceptable score. This situation arises when the shorter string is a sub-string of the longer string. However, we believe that with some modification, it can be made more efficient. In this paper, we presented a modified version of the Positional algorithm and the experimental results are satisfactory enough.

As noted above, traditional data flow is one-way from databases to data mining tools. The loop closes as data-mining insights are used to develop hypotheses that suggest new experiments. In this traditional view, simulation would be considered as just another data mining tool, to be called from the data-mining suite in order to assess correlations. However, realistic simulation differs from other tools in that it provides causal explanations rather than simple correlations. For this reason, the simulator is intercalated in the hypothesis return loop.

Using the simulator to control embedded database and data mining software facilitates the use of simulation as a focusing tool for knowledge discovery as shown in Figure 1. Several considerations suggest such a central role for neural simulation. First, there is the practical consideration of data quantity. Neural simulation, as it becomes more and more realistic, must become a massive consumer of experimental data, requiring a reliable pipeline. In addition to being dependent on experiment, simulation is itself an experimental pursuit, differing in this respect from the closed-form analytic models of traditional physics.

Once set-up, it can be hard to visualize the resulting system in order to verify that the system has been organized as planned. Storing parameters in a database is a valuable adjunct for checking and visualizing parameter sets and for storing and restoring them between simulations. Data mining can then be used both for comparing parameters among sets of simulations and for relating changes in parameter sets to changes in model dynamics. As compared to many data mining tools, neural simulation tends to be very computationally intensive, particularly when large parameter searches are undertaken. (Machine learning and artificial neural network algorithms are also computationally intensive.) Providing database and data mining tools within the simulator allows partial data analysis to be done at run-time.

3. A functionality Statement (FS)

FS is implemented as a series of low-level routines that operates on vectors that are maintained as the parallel columns of a table. FS thereby provides access to a variety of vector (array) manipulation tools built into NEURON. These vector functions permit convolution, numerical differentiation and integration, and basic statistics. Additional data mining tools have been added on by compiling C-language code that can be directly applied to the numerical vectors used as columns for a table. Vector-oriented C-language code is readily available from a variety of sources.[22,23] Such code can be compiled into NEURON after adding brief linking headers.

FS handles basic data basing functionality including: 1. creating tables; 2. inserting, deleting and altering tuples; 3. data queries. More sophisticated data basing functionality such as indexing and transaction protection are not yet implemented. Data basing commands in FS provide

1. Selection of specified data with both numerical and limited string criteria;

2. Numerical sorting;

3. Printing of data-slices by column and row designators;

4. Line, bar and scatter graphs;

5. Import and export of data in columnar format; 6. Symbolic spreadsheet functionality;

7. Iterators over data subsets or over an entire table;

8. Relational selections using criteria across related tables;

9. Mapping of user specified functions onto particular columns.

Among basic database functions, querying is the most complex. A query language, although often regarded as a database component and thereby denigrated as a data-mining tool, is a critical aspect of data-mining. Structured Query Language (SQL), because of its commercial antecedents, is less numerically oriented than is desirable for scientific query. The FS select command is designed to focus on

International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.1,No.1, 2011
DOI: 10.7321/jscse.v1.n1.3

numerical comparisons. Due to the importance of geometric information in neuroscience, inclusion of geometric criteria will be an additional feature that would be desirable in further development of FS.

The FS Select () command is similar to the commands related to the WHERE and HAVING

Sub-functions of SQL's SELECT. FS syntax naturally differs from that of SQL as it must follow the syntax of NEURON's object-oriented hoc language. An FS database table is a template in hoc.

The FS Select () command takes any number of arguments in sets. Each set consists of a column name, a comparative operator such as '<' or '==' and one or two arguments depending on the operator. Multiple criteria in a single Select () statement are handled with an implicit AND. A flag can be set to use OR on the clauses. A command can also begin with "&&" or "||" to return the union or respectively intersection of the selected rows with previously selected rows (*cf.* SQL UNION, INTERSECT subcommands). Although the FS Select () does not replicate the agglutinative syntax of SQL SELECT, this functionality can be effected by serial application of FS's Select (), sort() and stat () functions. Inner join between tables is implemented to permit formation of relational databases consisting of multiple tables.

4. Conclusion:

This paper studies the data mining approach can be use to process of dynamic behavior of a biochemical network from different perspectives. Most bioinformatics tools require specialized input formats for sequence comparison and analysis. This is particularly true for molecular phylogeny programs, which accept only certain formats. In addition, it is often necessary to eliminate highly similar sequences among the input, especially when the dataset is large. Moreover, most programs have restrictions upon the sequence name. We designed an algorithm SQL that works well with less information and assumptions. In particular, FL SQL is sequence order independent and can perform subset alignment with a more restrictive similarity measurement. Our proposed algorithm FL SQL makes use of the geometric hashing technique from computer vision, and the frequent item set mining technique from data mining. Both techniques have been used in a wide range of applications and algorithms. We demonstrated the efficiency and effectiveness of the SQL algorithm through experiments. SQL compares structures simultaneously, discovers large common substructures, and ensures that the common substructures detected are similar to each other.

References

[1]. Imielinski T, Mannila H. A database perspective on knowledge discovery. Communications of the ACM 1996;39:58–64.
[2]. Han J, Altman R, Kumar V, Mannila H, Pregibon D. Emerging scientific applications in data mining. Communications of the ACM 2002;45:54–58.
[3]. Churchland, P.; Sejnowski, T. The Computational Brain. MIT Press; 1994.
[4]. Hand D. Statistics and data mining: Intersecting disciplines. ACM SIGKDD 1999;1:16–19.
[5]. Hirji K. Exploring data mining implementation. Communications of the ACM 2001;44:87–93.
[6]. Wei G, Liu D, Liang C. Charting gene regulatory networks: strategies, challenges and perspectives. Biochemical Journal 2004;381:1–12.
[7]. Winslow R, Boguski M. Genome informatics: current status and future prospects. Circulation Research 2003;92:953–961.
[8]. Concha M, Adams R. Oriented cell divisions and cellular morphogenesis in the zebrafish gastrula and neurula: a time-lapse analysis. Development 1998;125:983–994.
[9]. Martone M, Zhang S, Gupta A, Qian X, He H, Price D, Wong M, Santini S, Ellisman M. The cellcentered database: a database for multiscale structural and protein localization data from light and electron microscopy. Neuroinformatics 2003;1:379–395.

[10]. Stephan K, Kamper L, Bozkurt A, Burns G, Young M, Kotter R. Advanced database methodology for the collation of connectivity data on the macaque brain (cocomac). Phil. Trans. R. Soc. Lond. B 2001;356:1159–1186.
[11]. Senft S, Ascoli G. Reconstruction of brain networks by algorithmic amplification of morphometry data. Lecture Notes Computer Science 1999;1606:25–33.
[12]. Langer S. Openrims: an open architecture radiology informatics management system. Journal of Digital Imaging 2002;15:91–97.
[13]. Megalooikonomou V, Ford J, Shen L, Makedon F, Saykin A. Data mining in brain imaging. Statistical Methods in Medical Research 2000;9:359–394.
[14]. Neill M, Hilgetag C. The portable UNIX programming system (PUPS) and CANTOR: a computational environment for dynamical representation and analysis of complex neurobiological data. Phil. Trans. R. Soc. Lond. B 2001;356:1259–1276.
[15]. Zhu J, Lytton W, Uhlrich D. An intrinsic oscillation in interneurons of the rat lateral geniculate nucleus. J Neurophysiol 1999;81:702–711.
[16]. Sekerli M, Negro C, Lee R, Butera R. Estimating action potential thresholds from neuronal timeseries: new metrics and evaluation of methodologies. IEEE Transactions on Biomedical Engineering 2004;51:1665–1672.
[17]. Hazarika N, Chen J, Tsoi A, Sergejew A. Classification of EEG signals using the wavelet transform. Signal Processing 1997;59:61–72.
[18]. Delorme A, Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods 2004;134:9–21.
[19]. Victor J, Purpura K. Metric-space analysis of spike trains - theory, algorithms and application. Network-Computation in Neural Systems 1997;8:127–164.
[20] E Schrodinger. Cambridge University Press: Cambridge, UK, 1944.
[21] R Giancarlo et al. A synopsis Bioinformatics 25:1575, 2009.
[22] EV Koonin. Bioinformatics 15: 265,1999.
[23] JC Wooley. J.Comput.Biol 6: 459, 1999.
[24] CH Bennett et al. IEEE Trans.Inform.Theory 44: 4, 1998.
[25] S Grumbach & F Tahi. Journal of Information Processing and Management 30(6): 875, 1994.
[26] E Rivals et al. A guaranteed compression scheme for repetitive DNA sequences. LIFL, Lille I University, technical report IT-285, 1995.
[27] X Chen et al. A compression algorithm for DNA sequences and its applications in Genome comparison. In Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, Tokyo, Japan, April 8-11, 2000.
[28] TC Bell et al. Newyork:Prentice Hall,1990.
[29] J Ziv & A Lempel. IEEE Trans. Inf. Theory 23: 337, 1977.
[30] J Ziv & A Lempel. IEEE Trans. Inf.Theory, 24: 530, 1978.
[31] M. L. Lee, T. W. Ling and W. L. Low "IntelliClean: A knowledge-based intelligent data cleaner," In *Proc. Sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, Boston, USA, pp. 290-294, August 2000.
[32] Y. R. Wang, S. E. Madnick, and D. C. Horton, "Interdatabase instance identification in composite information systems," In *Proc. Twenty-Second Annual Hawaii International Conference on System Sciences,* Kailua- Kona, pp. 677–684, January 1989.
[33] A. Monge, "Matching algorithms within a duplicate detection system," *IEEE Data Engineering Bulletin,* vol. 23, no. 4, pp.14-20, 2000.
[34] M. A. Hernandez and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the Merge/Purge problem," *Data Mining and Knowledge Discovery,* vol. 2, no. 1, pp. 9-37, 1998.
[35] A. E. Monge and C. Elkan. "An efficient domain independent algorithm for detecting approximately duplicate database records," In *Proc. SIGMOD Workshop on Data Mining and Knowledge Discovery,* Arizona, pp. 23-29, May 1997.
[36] R. Ananthakrishna, S. Chaudhuri and V. Ganti, "Eliminating fuzzy duplicates in data warehouses," In *Proc. 28th VLDB Conference, Hong Kong, China,* pp. 586- 597, 2002.
[37] R. S. Boyer and J. S. Moore, "A fast string-searching algorithm," *Communications of the ACM,* vol. 20, no. 10, pp. 762–772, 1977.