Mining Substructures in Protein Data

Fedja Hadzic[1], Tharam S. Dillon[1], Amandeep S. Sidhu[1], Elizabeth Chang[2] and Henry Tan[1]

[1]*Faculty of Information Technology, University of Technology Sydney, Australia*
E-mail: (fhadzic, tharam, asidhu, henryws)@it.uts.edu.au
[2]*School of Information Systems, Curtin University of Technology Perth, Australia*
E-mail: Elizabeth.Chang@cbs.curtin.edu.au

Abstract

In this paper we consider the 'Prions' database that describes protein instances stored for Human Prion Proteins. The Prions database can be viewed as a database of rooted ordered labeled subtrees. Mining frequent substructures from tree databases is an important task and it has gained a considerable amount of interest in areas such as XML mining, Bioinformatics, Web mining etc. This has given rise to the development of many tree mining algorithms which can aid in structural comparisons, association rule discovery and in general mining of tree structured knowledge representations. Previously we have developed the MB3 tree mining algorithm, which given a minimum support threshold, efficiently discovers all frequent embedded subtrees from a database of rooted ordered labeled subtrees. In this work we apply the algorithm to the Prions database in order to extract the frequently occurring patterns, which in this case are of induced subtree type. Obtaining the set of frequent induced subtrees from the Prions database can potentially reveal some useful knowledge. This aspect will be demonstrated by providing an analysis of the extracted frequent subtrees with respect to discovering interesting protein information. Furthermore, the minimum support threshold can be used as the controlling factor for answering specific queries posed on the Prions dataset. This approach is shown to be a viable technique for mining protein data.

Keywords

Protein discovery, association mining, frequent subtree mining, structure matching

1. Introduction

Data mining or knowledge discovery from data (KDD), in its most fundamental form, is to extract interesting, nontrivial, implicit, previously unknown and potentially useful information from data [1, 2]. In bioinformatics, this process could refer to summarizing rules for multiple DNA or protein sequences, finding motifs in sequences to predict folding patterns, discovering genetic mechanisms underlying a disease, and so on. With substantial growth of biological data, KDD is playing a more significant role in analyzing the data and in solving emerging problems. A critical problem in biological data analysis is to classify biological sequences and structures based on their critical features and functions. For example, gene sequences isolated from diseased and healthy tissues can be compared to identify critical differences between the two of the classes of genes. Such features can be used for classifying biological data and predicting behaviors. A lot of methods have been developed for biological data classification [3].

All biological experiments are driven by a plethora of experimental design hypotheses to be proven or rejected based on data values stored in multiple distributed biomedical databases, for example, genome or proteome databases. To extract and analyze the data perhaps poses a much bigger challenge for researchers than to generate the data [4]. To extract and analyze information from distributed biomedical databases, distributed heterogeneous data must be gathered, characterized and cleaned. However, domain specific ontologies such as Gene Ontology [5], MeSH [6] and Protein Ontology [7, 8, 9] exist to provide context and semantics to distributed biomedical data.

Frequent pattern analysis has been a focused theme of study in data mining, and a lot of algorithms and methods have been developed for mining frequent patterns, sequential patterns and structural patterns [2, 10]. However, not all the frequent pattern analysis methods can be adopted for analysis of complex biological data because many frequent pattern analysis methods are trying to discover perfect patterns, whereas most biological data patterns contain a substantial amount of noise or faults. For example, a DNA sequential pattern usually allows a nontrivial number of insertions, deletions, and mutations. Frequent sequential pattern discovery has been an active research area for years. Many algorithms have been developed and deployed for this purpose [11, 12, 13, 14]. One of the most popular pattern discovery algorithms for bioinformatics data is BLAST [15].

Besides finding sequential patterns, many biological data analysis tasks need to find frequent structured patterns, such as frequent protein or chemical compound structures from the data. In this paper we consider the 'Prions' database which describes a protein ontology instance store for Human Prion Proteins. XML format is used to store this data. The Prions database can be viewed as a database of rooted ordered labeled subtrees.

Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)
0-7695-2702-7/06 $20.00 © 2006 IEEE

Tree Mining has attracted lots of interest among the data mining community, due to the increasing use of semi-structured data sources for more meaningful knowledge representations. This is particularly evident in areas such as Bioinformatics, XML Mining, Web applications, scientific data management, and more generally in any area where the knowledge is represented in a tree-structured form. Many powerful tree mining algorithms have been developed to aid in structural comparisons, association rule discovery and in general, for mining of tree structured knowledge representations. The problem of frequent subtree mining can be generally stated as follows. Given a tree database T_{db} and minimum support threshold (σ), find all subtrees that occur at least σ times in T_{db} The two known types of subtrees are induced and embedded. An induced subtree is a subtree where the parent-child relationships must be the same to those in the original tree. In addition to this, an embedded subtree allows a parent in the subtree to be an ancestor in the original tree and hence the information about ancestor-descendant relationships is kept. Furthermore, two different support definitions used are occurrence-match and transactional support. Occurrence-match support takes repetition of items within a transaction into account, while the transaction based support only checks for the existence of the items in a transaction.

Our work in the area of frequent subtree mining is characterized by adopting a Tree Model Guided (TMG) candidate generation [16, 17] as opposed to the join approach which is commonly used. This non-redundant systematic enumeration model ensures only valid candidates are generated which conform to the actual tree structure of the data. Furthermore, our unique Embedding List representation of the tree structure has allowed for an efficient implementation of the TMG approach which has resulted in efficient algorithms for mining embedded (MB3) [16] and induced (IMB3) [18] subtrees, from a databases of labeled rooted ordered subtrees. $MB3^{-R}$ and $IMB3^{-R}$ algorithms [19] are latest implementations that adopt a more space efficient global representation and only store the right most path information for candidate subtrees.

In this paper we apply the $MB3^{-R}$ algorithm to the Prions database in order to extract the frequently occurring subtrees. Since the maximum depth of the subtree present in the Prions database is equal to one, all the extracted subtrees will be of induced type. Different support thresholds will be used and the extracted patterns will be accompanied with an in-depth analysis which explains the higher potential of the method for the discovery of useful protein information.

The rest of the paper is organized as follows. Section 2 briefly defines the problem of frequent subtree mining. The related works in the area of tree mining and some applications in bioinformatics are overviewed in Section 3. An overview of the $MB3^{-R}$ algorithm is provided in Section 4. In section 5, we apply the algorithm to the Prions dataset, and a biological explanation of the results is given in Section 6. Section 7 concludes the paper.

2. Problem definition

This section provides a general definition of the problem of frequent subtree mining. Due to the space limitations and the current scope of our work, we do not provide a detailed overview of the basic tree concepts, but refer the reader to our previous works [16, 17, 19], where such detailed overview has been provided.

Mining frequent subtrees. Let T_{db} be a tree database consisting of N transactions of trees, K_N. Given a minimum support threshold (σ), the task of frequent subtree mining is to find all the candidate embedded subtrees that occur at least σ times in T_{db}.

3. Related Works

Tree mining algorithms are increasingly being developed and the scope of their application usually depends on the assumptions made about the tree structure that the algorithm can be applied to. Naturally, these assumptions depend upon the domain of interest, where the developed algorithm is to be applied.

Hence, many tree mining algorithms exist and they can be distinguished based upon the types of tree patterns that they extract. PathJoin [20], uNot [21], uFreqt [22], and HybridTreeMiner [23], mine induced, unordered trees. AMOT [24], mines induced ordered trees, and by using 'right-and-left tree join' method it efficiently enumerates only those candidates that have a high probability of being frequent. Treeminer [25], is an efficient algorithm for discovering all frequent embedded subtrees in a forest using a data structure called the vertical scope-list. Zaki has developed an efficient algorithm for mining frequent embedded unordered subtrees, SLEUTH [26], to be applied to the cases when the information about the order of the sibling nodes in the data tree is not important or available. TreeFinder [27] uses an Inductive Logic Programming approach to mine unordered, embedded subtrees, but in the process many frequent subtrees can be missed. In regards to the application of tree mining to biological data, some approaches have been developed for analysis of phylogenetic databases [28, 29].

Besides the tree mining work there have been many recent developments in graph mining. Apriori-based Graph Mining (AGM) approach was introduced in [30] and it mines induced subgraphs which can be disconnected. FSG algorithm [31], guarantees that the extracted subgraphs are connected. Warmr algorithm is a level-wise Inductive Logic Programming based technique, used in [32] for discovering frequent substructures of chemical compounds in relation to their possible carcinogenicity. FreeTreeMiner [33] extracts unrooted unordered trees from a graph

database. Yan and Han in 2002 [34] have introduced a lexicographical ordering system among graphs, based upon which the gSpan algorithm uses a depth first search strategy to mine frequent connected subgraphs. Heymans, and Singh in 2003 [35] have presented a graph comparison algorithm for computing the evolutionary distance between two metabolic pathways, useful for phylogenetic analysis.

4. MB3^{-R} algorithm

Due to the space limitations and the scope of this work, this section only provides a brief overview of the MB3^{-R} algorithm. For a more detailed description please refer to [19]. Those interested in obtaining the source code of the algorithm, should feel free to email the authors.

The database of XML documents is first transformed into a database of rooted integer-labeled ordered tree. The tree database is traversed once to create a global sequence which stores each node in the pre-order traversal together with the necessary node information. At the same time the set of frequent 1-subtrees is obtained by hashing the encountered node labels. Once the databases is traversed the global sequence is used to construct the Recursive List (RL) [19]. Thereafter, the TMG candidate generation using the RL structure takes place and for each k ≥1 the RMP coordinates of each frequent (k-1)-subtree are stored in 'Fk-1' hashtable. Before a subtree is stored in the frequent hashtable, full k-1 pruning [16, 18] is performed to ensure that all its subtrees are also frequent. Each frequent (k-1)-subtree is extended one node at a time, starting from the last node of its RMP (right most node), up to its root, whereby all k-subtrees are enumerated. The whole process is repeated until all k-subtrees are enumerated and counted.

5. MB3^{-R} application to Prions database

Prion (short for proteinaceous infectious particle) is a type of infectious agent. Prions are abnormally structured forms of host protein, which are able to convert normal molecules of protein into abnormally structured form. Prions dataset describes Protein Ontology [7] database for Human Prion proteins in XML format [9]. The XML tags are first mapped to integer indexes similar to the format used in TMGJ [16] and [25]. Representing label as integer instead of a string label has considerable performance and space advantages [16]. Since the maximum height of the Prions tree structure is 1, all candidate subtrees generated are induced subtrees. The experiments were run on 3Ghz (Intel-CPU), 2Gb RAM, Mandrake 10.2 Linux machine and compilation was performed using GNU g++ (3.4.3) with –g and –O3 parameters. Occurrence-match support definition was used. The total run-time and memory usage of the MB3 algorithm is displayed in Fig. 1, for varying support thresholds. The next section provides an analysis and explanation of some of the extracted frequent patterns.

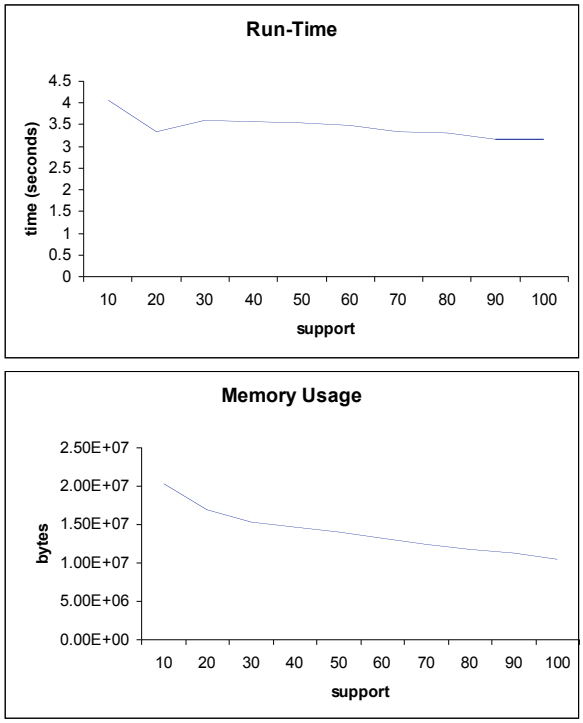

Fig. 1 MB3^{-R} run-time and memory usage profile

6. Biological Interpretation

In bioinformatics, the discovery of structural patterns by matching data representation structures is essential for analysis and understanding of biological data. If a structural pattern occurs frequently, it is ought to be important in some way. On the other hand, infrequent patterns may also provide meaningful information. Thus to extract meaningful information from biological data we need to mine both sequence and structural patterns. In this section show how patterns discovered in prion dataset by the MB3-R algorithm, when applied to the Prions dataset help in extracting meaningful information.

ATOMSequence

_ATOM_Chain[L]

_ATOM_Residue[ALA]

This pattern was discovered 40 times in the dataset. Here ATOMSequence contains refers to chain of residue sequence for the atom list in question (_ATOM_Chain = L). Description of the structure of the Chain refers to numerous instances of Residues defined like (_ATOM_Residue = ALA). Each residue has a number of Atoms linked to it, which make up the atom list for the residue. Similarly collection of atom lists for all residues in the chain describes the entire ATOMSequence.

ATOMSequence

_ATOM_Chain[A]

_ATOM_Residue[ALA]

Atom[H]

This pattern was discovered 101 times in the data. The pattern is a similar extension of the pattern discussed above, with the inclusion of identifying the Atom (Atom = H) linked to the residue.

ATOMSequence

_ATOM_Chain[H]

_ATOM_Residue[THR]

Atom[CA]

Occupancy[1]

Element[C]

The pattern shown above is discovered 100 times in the data. This pattern identifies more details about the Atom linked to the Residue (like: Occupancy = 1 and Element = C).

These are some the patterns discovered by the patterns discovered by the MB3^{-R} algorithm. The algorithm aids in discovering useful pattern structures in Protein Ontology datasets, which makes it useful for comparison of protein datasets taken across protein families and species and helps in discovering interesting similarities and differences.

7. Conclusions & future work

This paper has presented the application of the MB3^{-R} tree mining algorithm to the tree structured Prions protein database. The aim was to extract the frequently occurring subtrees which have the potential of providing useful information and knowledge related to these proteins. The experiments were accompanied with a biological interpretation of some interesting patterns that were discovered. This indicates the potential of the tree mining algorithms providing interesting biological information when applied to tree structured biological data. Graph and Tree structured data is increasingly in use for many representations of biological knowledge and hence some of our future work will involve the development of graph mining algorithms that can efficiently extract frequently occurring sub-graphs from a large graph structured database. We also intend to use these tree mining algorithms to examine other classes of protein datasets, and to fine tune their use for such data.

8. References

[1] J. T. L. Wang, M. J. Zaki, H. T. T. Toivonen, and D. Shasha, "Data Mining in Bioinformatics," in Advanced Information and Knowledge Processing, X. Wu and L. Jain, Eds. London: Springer, 2005.

[2] J. Han and M. Kamber, Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann, 2001.

[3] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction. New York: Springer-Verlag, 2001.

[4] A. Holloway, R. K. van Laar, R. W. Tothill, and D. Bowtell, "Options available - from start to finish - for obtaining data from DNA microarrays II," Nature Genetics Supplement, vol. 32, pp. 481-489, 2002.

[5] M. Ashburner, C. A. Ball, J. A. Blake, H. Butler, J. C. Cherry, J. Corradi, and K. Dolinski, "Creating the Gene Ontology Resource: Design and Implementation," Genome Research, vol. 11, pp. 1425-1433, 2001.

[6] S. J. Nelson, D. Johnston, and B. L. Humphreys, "Relationships in Medical Subject Headings," in Relationships in the organization of knowledge, C. A. Bean and R. Green, Eds. New York: Kluwer Academic Publishers, 2001, pp. 171-184.

[7] A. S. Sidhu, T. S. Dillon, and E. Chang, "Protein Ontology," in Database Modeling in Biology: Practices and Challenges, Z. Ma and J. Y. Chen, Eds. New York: Springer, 2006, pp. 39-60.

[8] A. S. Sidhu, T. S. Dillon, E. Chang, and B. S. Sidhu, "Protein ontology: vocabulary for protein data," presented at 3rd International IEEE Conference on Information Technology and Applications, 2005 (IEEE ICITA 2005), Sydney, 2005.

[9] A. S. Sidhu, T. S. Dillon, B. S. Sidhu, and H. Setiawan, "A Unified Representation of Protein Structure Databases," in Biotechnological Approaches for Sustainable Development, M. S. Reddy and S. Khanna, Eds. India: Allied Publishers, 2004, pp. 396-408.

[10] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules " presented at International Conference of Very Large Data Bases, Santiago, Chile, 1994.

[11] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth", In ICDE'01, 2001, pp. 215–226.

[12] H. Tan, T.S. Dillon, F. Hadzic, and E. Chang, "SEQUEST: mining frequent subsequences using DMA Strips", in Data Mining and Information Engineering 2006, Prague, Czech Republic, 2006.

[13] Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning (2000), 1-31.

[14] R. Agrawal, R. Srikant, "Mining Sequential Patterns", In Proc. IEEE 11th ICDE, Vol. 6, No. 10, 1995, pp. 3–14.

[15] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," Journal of Molecular Biology, vol. 215, pp. 403-410, 1990.

[16] H. Tan, T.S. Dillon, F. Hadzic, E. Chang, and L. Feng, "MB3 Miner: mining eMBedded sub-TREEs using Tree Model Guided candidate generation", In Proc. of the 1st International Workshop on Mining Complex Data, held in conjunction with ICDM'05, Houston, Texas, USA, 2005.

[17] H. Tan, T.S. Dillon, F. Hadzic, L. Feng, and E. Chang, "Tree Model Guided Candidate Generation for Mining Frequent

Subtrees from XML", Submitted to Transactions on Knowledge Discovery from Data (TKDD), January, 2006.
[18] H. Tan, T.S. Dillon, F. Hadzic, L. Feng, and E. Chang, "IMB3 Miner: Mining Induced/Embedded Subtrees by Constraining the Level of Embedding", In Proc. of PAKDD'06, Singapore, 2006.
[19] H. Tan, T.S. Dillon, F. Hadzic, E. Chang, and L. Feng, "Mining induced/embedded subtrees using the level of embedding constraint", Submitted to Knowledge and Information Systems An International Journal, Springer, 2006.
[20] Y. Xiao, J.-F. Yao, Z. Li, and M.H. Dunham, "Efficient data mining for maximal frequent subtrees", In Proc. of the 3rd IEEE International Conference on Data Mining (ICDM 2003), Melbourne, Florida, USA, 2003, pp. 379-386.
[21] T. Asai, H. Arimura, T. Uno, S. Nakano, "Discovering Frequent Substructures in Large Unordered Trees", The 6th International Conference on Discovery Science, 2003.
[22] S. Nijssen, J.N. Kok, "Efficient discovery of frequent unordered trees", In Proc. of the 1st International Workshop Mining Graphs, Trees, and Sequences (MGTS-2003), Dubrovnik, Croatia, 2003.
[23] Y. Chi, Y. Yang, and R.R. Muntz, "HybridTreeMiner: An efficient algorihtm for mining frequent rooted trees and free trees using canonical forms", In Proc. of the 16th International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 2004.
[24] S. Hido and H. Kawano, H. "AMIOT: Induced Ordered Tree Mining in Tree-structured Databases", In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, Texas, USA, 2005, pp 170-177.
[25] M.J. Zaki. Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. In IEEE Transaction on Knowledge and Data Engineering, 17, 8, 2005, 1021-1035.
[26] M.J. Zaki, "Efficiently Mining Frequent Embedded Unordered Trees", Fundamenta Informaticae 65, IOS Press, 2005, pp. 1-20.
[27] A. Termier, M-C. Rousset, and M. Sebag, "Treefinder: A First Step Towards XML Data Mining" In Proc. of IEEE ICDM'02, 2002.
[28] D. Shasha, J.T.L. Wang, S. Zhang, "Unordered Tree Mining with Applications to Phylogeny", 20th International Conference on Data Engineering, 2004.
[29] J. T. L. Wang, H. Shan, D. Shasha, and W. H. Piel, "Treerank: A similarity measure for nearest neighbor searching in phylogenetic databases", In Proc. of the 15th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'03), 2003.
[30] A. Inokuchi, T. Washio, and H. Motoda, "An Apriori-based algorithm for mining frequent substructures from graph data", In D.A. Zighed, H.J.Komorowski, and J.M. Zytkow, editors, Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2000 pp. 13–23.
[31] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs. IEEE Transactions Knowledge and Data Engineering, vol. 16, no. 9, 2004, pp. 1038-1051.
[32] L. Dehaspe, H. Toivonen, R. King, "Finding frequent substructures in chemical compounds", In 4th Intl. Conf. Knowledge Discovery and Data Mining, 1998.
[33] U. Ruckert and S. Kramer, "Frequent free tree discovery in graph data" In Proc. of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus, 2004, pp. 564 – 570.
[34] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining", In Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, 2002, pp. 721-724.
[35] M. Heymans, and A.K. Singh, "Deriving phylogenetic trees from the similarity analysis of metabolic pathways", Bioinformatics 19, 2003, pp. 138-146.