1A2-3

DryadeによるGene Networkデータからの飽和頻出DAGマイニング

Mining Closed Frequent DAGs from Gene Network Data with Dryade

ターミエ　アレックサンドル[*1] Alexandre Termier
鷲尾隆[*2] Takashi Washio
樋口知之[*1] Tomoyuki Higuchi
玉田嘉紀[*1] Yoshinori Tamada
井元清哉[*3] Seiya Imoto
大原剛三[*2] Kouzou Ohara
元田浩[*2] Hiroshi Motoda

[*1]統計数理研究所 Institute of Statistical Mathematics, Tokyo
[*2]大阪大学産業科学研究所 I.S.I.R., Osaka University
[*3]東京大学医科学研究所 Institute of Medical Science, University of Tokyo

We present in this article a new method to extract frequent patterns from gene networks. The particularity of this method is to be able to extract embedded sub-DAGs from the data, whereas previous methods were limited to extracting induced sub-DAGs. Our algorithm builds up upon our DRYADE closed frequent embedded attribute subtree mining algorithm, and by postprocessing its outputs discovers closed frequent embedded attribute sub-DAGs with one root in the data. We have tested our method on real gene networks data, and confirmed the existence of specific embedded sub-DAGs, that could not be found with previous algorithms limited to extracting induced sub-DAGs.

1. Introduction

Understanding the regulatory relationships of several genes is one of the key challenges of today's bioinformatics. Especially, there are still lots of genes whose functions are unknown. Discovering frequent interaction between these genes and genes whose function is known could help discover what the unknown functions are. Data-mining can help by providing automatic analysis of huge quantities of gene interaction data.

In this paper, we are interested in analyzing gene networks as described in [IGM02]. Each gene network has a structure of Directed Acyclic Graph (DAG), where the nodes are the genes. Extracting sub-DAGs patterns from DAGs is a complex task. There is only one specialized algorithm as of now, and else one has to resort to the even more computationnaly intensive graph-mining algorithms like [IWM03]. All the previoulsy mentionned algorithms can only mine induced sub-DAGs, which means that to find a pattern where gene A interacts with gene B, gene A and gene B must always have a direct interaction in the data, which makes the mining process very sensitive to noise. Moreover, the gene networks that we use as data are produced by a greedy algorithm, which heuristically tries to find an optimal network, but is not stable, so our data is noisy.

Hence, we propose a new method that is more noise-resistant and can discover more complex patterns, by discovering embedded sub-DAGs. This is performed by a modified version of our DRYADE closed frequent embedded subtree mining algorithm [TRS04]. By adapting it to DAGs, it can discover in DAG data closed frequent embedded sub-DAGs.

連絡先: ターミエ・アレックサンドル，統計数理研究所，〒106-8569　東京都港区南麻布 4-6-7，termier@ism.ac.jp

The paper is organised as follows: Section 2. describes in details our data and the problem at hand. Section 3. describes briefly the Dryade algorithm, and the modifications necessary to handle DAGs. Section 4. reports on our first experiments. Last, Section 5. concludes the paper and gives some perspectives for future research.

2. Preliminaries

2.1 Formal definitions

A **labelled graph** is a tuple $G = (N, E, \varphi)$, where N is the set of nodes, $E \subseteq N \times N$ is the set of edges, and $\varphi : N \mapsto L$ is a labelling function with L a finite set of labels. For an edge $(u, v) \in E$, u is the **parent** of v and v is the **child** of u. If there is a set of nodes $\{u_1, ..., u_n\} \subseteq N$ such that $(u_1, u_2) \in E$,...,$(u_{n-1}, u_n) \in E$, $\{u_1, ..., u_n\}$ is called a **path**, u_1 is an **ancestor** of u_n and u_n is a **descendant** of u_1. There is a **cycle** in the graph if a path can be found from a node to itself. A **labelled DAG** is a labelled graph without cycles. A **labelled tree** is a labelled DAG where each node can only have one parent, except one node which has no parents and which is called the **root**. An **attribute DAG (*tree*)** is a DAG (*tree*) where there cannot be two siblings with the same label. In this paper we only deal with attribute DAGs and attribute trees, for now on we will simply refer attribute DAGs by "DAGs" and attribute trees by "trees" by abuse of notation.

Let $P_1 = (N_1, E_1, \varphi_1)$ and $P_2 = (N_2, E_2, \varphi_2)$ be two DAGs. P_1 is an **embedded** sub-DAG of P_2 if there exists an injective isomorphism $\mu : P_1 \mapsto P_2$ such as: 1) for two nodes $u \in N_1$ and $v \in N_2$ such that $v = \mu(u)$, $\varphi_2(v) = \varphi_1(u)$ holds and 2) for $(u, v) \in E_1$, $\mu(u)$ is an ancestor of $\mu(v)$ in P_2. In short, the isomorphism must preserve the labels and the ancestor relationship. If in 2), the isomorphism only preserves the parent relationship, then

P_1 is an **induced** sub-DAG of P_2. Note that an induced sub-DAG is also an embedded sub-DAG, but not the opposite.

Let $\mathcal{D} = \{D_1, ..., D_n\}$ be a set of labelled DAGs and $\varepsilon \geq 0$ be an absolute frequency threshold. A DAG P is a **frequent embedded (*induced*)** sub-DAG of $\mathcal{D}$ if it is embedded (*induced*) in at least ε DAGs of $\mathcal{D}$. The set of DAGs of $\mathcal{D}$ in which P is embedded (*induced*) is called the **support** of P, denoted $support(P)$.

A frequent embedded (*induced*) sub-DAG P of $\mathcal{D}$ is **closed** if it is maximal for its support, i.e. if there is no frequent embedded (*induced*) sub-DAG P' of $\mathcal{D}$ such that P is embedded (*induced*) into P' and $support(P) = support(P')$.

2.2 Problem definition

Our input data are gene networks which have a structure of labelled DAG, where the labels of the nodes are the names of the genes, and there is an edge from a node labelled with gene A to node labelled with gene B if there is a direct interaction between gene A and gene B.

The problem we are interested in is to find all the closed frequent embedded sub-DAGs from the DAG-structured gene networks described above. For these preliminary works, we limit the closed frequent embedded sub-DAGs that must be discovered to the closed frequent embedded sub-DAGs with only one root.

3. Mining closed frequent sub-DAGs

To our knowledge, there is no algorithm capable of extracting embedded sub-DAG patterns as we define them. The only algorithm specifically designed to mine frequent sub-DAGs has been presented in [CKK04], but it is limited to finding induced sub-DAGs. One possibility could be to use a graph-mining algorithm like AGM [IWM03] instead, but current graph-mining algorithms cannot mine embedded subgraphs.

We propose an extension of our DRYADE algorithm. DRYADE is able to extract all the closed frequent embedded attribute sub-trees from a collection of tree-structured data.

The modifications made to the algorithm were twofold. First, the input processing part was changed in order to accept DAG input. This was a quite straighforward modification: we only had to change the internal representation of input data to handle DAGs. Because of the similarities between trees and DAGs, the rest of the algorithm can stay unchanged and discover correctly closed frequent sub-trees from this DAG input.

Then, a post-processor was added after the tree-mining step, which analyses the closed frequent sub-trees in order to discover the closed frequent sub-DAGs with one root.

The post-processor exploits the fact that DRYADE computes all the closed frequent sub-trees as well as all their mappings in the data. Hence, for a given closed frequent sub-tree, its actual mappings in the data can be examined to assess if the mapped data corresponds to a tree or a DAG structure. For each mapping its actual structure is recorded, and the structures that are not frequent are discarded.

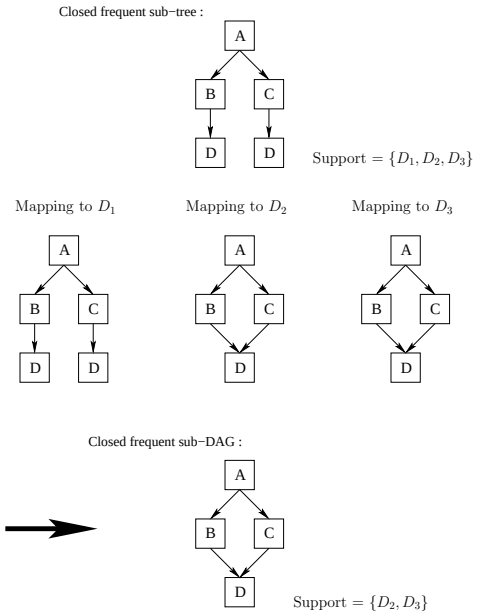

図 1: Finding closed frequent sub-DAGs from closed frequent sub-tree mappings, for $\varepsilon = 2$

For example consider the closed frequent sub-tree of Fig. 1. It is supported by 3 DAGs, but appears as a tree in D_1, and appears as a DAG in D_2 and D_3. The tree shape appearing only once, it is in fact not frequent w.r.t. a minimal frequency threshold of 2, so the output of the postprocessor is the DAG shape supported by two DAGs of the data.

To ease the task of the bioinformaticians analysing the output sub-DAGs of DRYADE, we added a filter that prunes the closed frequent sub-DAGs of low interest. These sub-DAGs are the sub-DAGs which are in facts induced sub-DAGs from the data. This means that all the parent-child edges in the closed frequent sub-DAG are also direct parent-child edges in the data. Such patterns are "obvious" and make very little abstraction over the data, so the bioinformaticians are less interested in seeing them.

4. Experiments

We have performed preliminary experiments with real biological data. The source of the gene networks we are interested in is raw microarray data from [SSZ+98]. This raw data is processed with the method proposed in [IGM02] to produce the DAG-structured gene networks.

There are 5,000 DAGs, each having 801 nodes. With an absolute support threshold of 2,500 (50%), our algorithm found a total of 543 DAG patterns. 209 of these patterns passed the pruning step described above and were reported as potentially interesting for bioinformaticians. Analysis of these 209 patterns by bioinformaticiens is ongoing.

We show one of the discovered patterns in Fig. 2. 3 out of the 5 genes of the pattern are involved in the lipid metabolism (YOL101C, YGL055W, YKL182W), according to the Gene Ontology [Con00]. However, the functions of

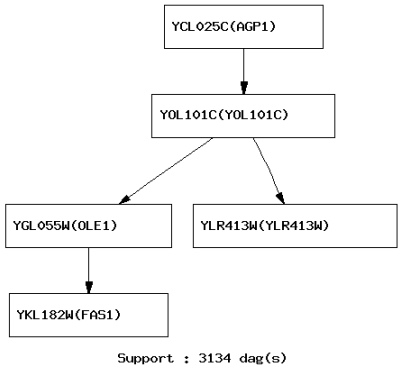

図 2: A discovered pattern

YLR413W are marqued as “unknown” in Gene Ontology. The pattern that our system found suggests that YLR413W could also be involved in the lipid metabolism.

5. Conclusion and future works

In this paper, we presented an algorithm for mining closed frequent embedded attribute one-root sub-DAG patterns from a collection of DAG data. As far as we know, this is the first attempt to mine embedded sub-DAG patterns. Our algorithm has been tested on real data, and could discover patterns that would not be found by classical methods focusing only on induced patterns. We hope that the analysis of these patterns will provide useful biological results.

参考文献

[CKK04] Yen-Liang Chen, Hung-Pin Kao, and Ming-Tat Ko. Mining dag patterns from dag databases. In *Web-Age Information Management (WAIM), Dalian, China*, 2004.

[Con00] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[IGM02] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian network and nonparametric regression. In *Pacific Symposium on Biocomputing*, pages 175–186, 2002.

[IWM03] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning*, 50(3):321–354, 2003.

[SSZ+98] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstien, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[TRS04] A. Termier, M.C. Rousset, and M. Sebag. Dryade : a new approach for discovering closed frequent trees in heterogeneous tree databases. In *International Conference on Data Mining ICDM'04, Brighton, England*, pages 543–546, 2004.