Extraction of Correlated Gene Clusters from Multiple Graph Structures: Theory

Akihiro Nakaya nakaya@kuicr.kyoto-u.ac.jp
Susumu Goto goto@kuicr.kyoto-u.ac.jp
Minoru Kanehisa kanehisa@kuicr.kyoto-u.ac.jp
Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan

Keywords: correlated gene cluster, binary relationship, subgraph isomorphism, clustering

1 Introduction

Correlated gene cluster. With respect to an organism whose genome sequence has been determined, we can totally order all the genes along the genome sequence. Besides such *geometrical* relationships among genes, *similarity* relationships based on ORF sequences or 3D structures, *functional* relationships with respect to a metabolic/regulatory pathway or degree of co-expression, and so on can be denoted by using a set of *binary relationships* in a general manner. Each set of binary relationships forms a graph structure called an *adjacency graph* of genes as a whole. Fig. 1 shows three adjacency graphs G_1 (genome), G_2 (pathway), and G_3 (co-expression). Each graph node corresponds to a gene, and two nodes are connected by an edge (expressed by a solid line) when they are related by the binary relationship. When a set of such adjacency graphs is given, finding a set of genes is attracting interest where all or most of the genes reserve their mutual relationships in multiple adjacency graphs (e.g., the light gray nodes and the dark gray nodes in Fig. 1). We call such a set of genes a *correlated gene cluster.*

Hyperedges among genes. Finding correlated gene clusters can be formalized as a subgraph isomorphism problem and has been proved to be NP-complete. Therefore, some heuristics are required to cope with this problem. To extract a correlated gene cluster from two adjacency graphs, Ogata *et al.* [2] introduced a notion called FRECS (Functionally Related Enzyme Clusters). They also introduced a set of inter-graph links (between two nodes that correspond to the same gene), and searched similar subgraphs in two adjacency graphs so that the nodes of the subgraphs are connected by the inter-graph links. They found, for instance, that particular five *Escherichia coli* genes are next to each other in two adjacency graphs that correspond to the genome and the metabolic pathway.

We extend the power of their algorithm by increasing the number of adjacency graphs so that the additional graphs provide information of gene-gene interactions that cannot be found by just two graphs. Fig. 1 shows examples of correlated gene clusters (C_1 and C_2) in three adjacency graphs G_1, G_2, and G_3. Here, a dashed line links nodes each of which represents the position of a specific gene in each adjacency graph explained as above. We call this linkage a *hyperedge* and define the similarity between two hyperedges so that it can reflect the similarity (the inverse of the shortest path length) between their nodes in each adjacency graph. By gathering hyperedges based on the similarity, we can find a set of nodes that are tightly coupled in adjacency graphs, that is, a correlated gene cluster.

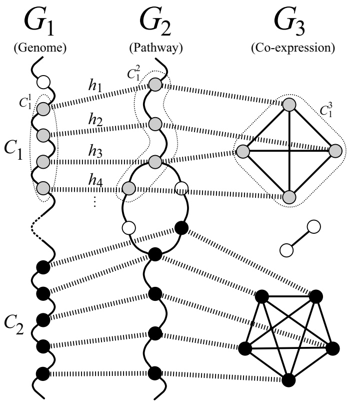

Figure 1: Gene cluster.

2 Method

Input dataset. As the input dataset, we use a set of adjacency graphs $G = \{G_1, \ldots, G_n\}$ and a set of hyperedges $H = \{h_1, \ldots, h_m\}$. When the number n of adjacency graphs are given as above, we denote a hyperedge with an n-tuple $h_i = (n_{1,i_1}, \ldots, n_{n,i_n})$. Here, the kth element $h_i^k = n_{k,i_k}$ is G_k's node that constitutes the hyperedge $(1 \leq k \leq n)$ and we assume that a hyperedge exactly consists of n nodes to make the problem simple.

Clustering of hyperedges. Suppose that there are n adjacency graphs. Let $C_1 = \{h_{n_1}, \ldots, h_{n_p}\}$ and $C_2 = \{h_{m_1}, \ldots, h_{m_q}\}$ be sets of hyperedges, and let $C_1^k = \{h_{n_1}^k, \ldots, h_{n_p}^k\}$ and $C_2^k = \{h_{m_1}^k, \ldots, h_{m_q}^k\}$ be sets of the kth nodes of hyperedges in C_1 and C_2, respectively. We define the dissimilarity between two sets of hyperedges C_1 and C_2 as $D(C_1, C_2) = \sum_{1 \leq s \leq n} dis(C_1^s, C_2^s)$. $dis(C_1^s, C_2^s)$ is the dissimilarity between C_1^s and C_2^s. Here, for example, $dis(C_1^s, C_2^s)$ is defined as $min\{d(x, y) | x \in C_1^s, y \in C_2^s\}$, where $d(x, y)$ is the length of the shortest path between nodes x and y in adjacency graph G_s (which can be calculated by Dijkstra's algorithm or Warshall-Floyd's algorithm).

Let C be the initial set of clusters each of which consists of a single hyperedge, i.e., $C = \{\{h_1\}, \ldots, \{h_m\}\}$. Starting with C, we iterate to pick two clusters between which dissimilarity is the smallest and merge them into a new cluster. To avoid distant genes being merged into the same cluster, we use a threshold defined for each adjacency graph. Let p_i be the threshold for adjacency graph G_i. When the path length between two nodes x and y is greater than p_i in G_i, we change the value of $d(x, y)$ to infinity, and leave the pairs of clusters whose dissimilarity is infinity untouched. When there are no cluster pairs whose dissimilarity is less than infinity we stop the clustering procedure and have correlated gene clusters.

3 Experimental Results and Discussions

We implemented the algorithm by using the C++ language on a Sun Microsystems Enterprise 10000 (Solaris 5.6) and a SiliconGraphics Origin 2000 (IRIX 6.5), and part of the program is parallelized by the POSIX thread library. We applied the algorithm to a series of datasets obtained from the KEGG database. It includes, for example, datasets of *Escherichia coli* (genome, metabolic pathway, and similarity of three-dimensional structure of enzyme) and *Synechocystis* sp. (genome, metabolic/regulatory pathway, and gene co-expression) and we found some correlated gene clusters. For details see Kawashima et al. [1].

As explained above, we have focused on only whether two genes are connected or not by means of a set of binary relationships. Now, we can extend the framework so that it can cope with weighted adjacency graphs according to the binding intensity of the relationships. Our algorithm works for this purpose, however, we must consider normalization of edge weights among different kinds of adjacency graphs (e.g., genome and pathway) since comparison between their absolute values do not always make sense.

Acknowledgements

Part of this work done by A.N. and S.G. was supported by Grant-in-Aid for Scientific Research on Priority Areas (C) "Genome Information Science" from the Ministry of Education, Science, Sport, and Culture of Japan, and computing resource was provided from the Supercomputer Laboratory, Kyoto University and Human Genome Center, University of Tokyo.

References

[1] Kawashima, S., Nakaya, A., Okuji, Y., Goto, S., and Kanehisa, M., Extraction of Correlated Gene Clusters from Multiple Graph Structures: Application, *Genome Informatics*, 11, 2000.

[2] Ogata, H., Fujibuchi, W., Goto, S., and Kanehisa, M., A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters, *Nucleic Acids Res.*, 28:4021–4028, 2000.