The Third International Symposium on Optimization and Systems Biology (OSB'09)
Zhangjiajie, China, September 20–22, 2009

Comparing Biological Networks via Graph Compression

Morihiro Hayashida[1,*] Tatsuya Akutsu[1,†]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University
Gokasho, Uji, Kyoto 611-0011, Japan

Abstract This paper proposes a novel method for comparing biological networks. In the proposed method, an original network structure is compressed by iteratively contracting identical edges. Then, the similarity of two networks is measured by a compression ratio of the concatenated networks. The proposed method is applied to comparison of metabolic networks of *H. sapiens*, *M. musculus*, *A. thaliana*, *D. melanogaster*, *C. elegans*, *E. coli*, *S. cerevisiae*, and *B. subtilis*. The results suggest that our method could efficiently measure the similarities between metabolic networks adequately.

Keywords Metabolic Networks; Graph Compression; Data Compression; Network Comparison; Morgan Index

1 Introduction

Comparison of various kinds of biological data is one of the central problems in bioinformatics and systems biology. Methods for comparison of DNA and/or protein sequences have been extensively studied and have been applied to analyses of real sequence data quite successfully. Due to increased interests in systems biology, extensive studies have recently been done on comparison of biological networks.

For comparison of metabolic networks, Ogata et al. developed a method based on clustering [12], Tohsato et al. extended a multiple sequence alignment technique to multiple alignment of metabolic pathways using a scoring scheme based on EC (Enzyme Commission) numbers [16], Pinter et al. applied a tree matching technique to alignment of metabolic pathways [14], and Wernicke and Rasche developed a simple backtracking algorithm utilizing the local diversity property [17]. For comparison of protein-protein interaction networks, Kelley et al. developed PathBlast using dynamic programming [5], Liang et al. developed NetAlign using a clique-based method for computing maximal common subgraphs [10], Li et al. developed MNAligner using integer quadratic programming [9], Singh et al. developed IsoRank algorithm based on Google's PageRank method [15], and Zaslavskiy et al. developed a gradient ascent-based method and a message passing-based method [19].

*Email: morihiro@kuicr.kyoto-u.ac.jp
†Email: takutsu@kuicr.kyoto-u.ac.jp

On the other hand, data compression methods have been applied to comparison of large sequence data [6, 8] and protein structure data [3, 7]. Since it is still difficult to compare global structures of large biological networks and data compression-based methods can be applied to comparison of large-scale sequence data, it is reasonable to try to apply data compression methods to comparison of biological networks. In this paper, we propose such a method.

In order to apply data compression to biological networks, data compression methods for graphs are required. For compression of graphs, Adler and Mitzenmacher developed a method based of Huffman coding of vertices [1], Peshkin developed GRAPHITOUR based on iterative contractions of identical edges [13], and Cook and Holder developed SUBDUE based on contraction of frequent subgraphs and MDL (minimum description length) principle [2], which was further extended to EDIF for lossless compression by Yang et al. [18] However, the method by Adler and Mitzenmacher does not seem to be useful for comparison of networks because it does not make much use of structural information. In GRAPHITOUR, the uniqueness of compression results is not guaranteed because there is some ambiguity in selection of overlapping edges (isomorphic graphs may be differently compressed depending on the orderings of vertices in input data), which is not suitable for comparison of network structures. This point is also unclear in EGIF and SUBDUE. Therefore, we develop in this paper a novel graph compression method for which it is guaranteed that two isomorphic graphs are compressed in the same way.

Using this compression method, we measure the similarity of two networks by means of the universal similarity metric (USM) proposed by Li *et al.* [8] USM is defined using Kolmogorov complexity which represents the amount of information contained in data, and is obtained by removing redundant parts maximally. Therefore, Kolmogorov complexities are approximated by compression sizes.

We apply the proposed method to comparison of metabolic networks. The results of hierarchical clustering for several organisms suggest that the proposed method could measure the similarities between metabolic networks adequately.

2 Graph Compression Method

Since our proposed method is based on GRAPHITOUR, we briefly review the GRAPHITOUR algorithm [13]. GRAPHITOUR is based on iterative contractions of identical edges. In order to efficiently contract edges, GRAPHITOUR selects edges appearing more frequently, and solves an instance of *maximum cardinality matching* problem, which finds as many edges as possible such that no two edges share a common vertex.

Fig. 1 shows an example of contraction of identical edges. The graph of (A) contains 4 edges labeled with 'a' and 'b', 2 edges with 'a' and 'a', 1 edge with 'b' and 'b', and 1 edge with 'a' and 'c'. GRAPHITOUR selects edges with 'a' and 'b' because they appear most frequently, and solves maximum cardinality matching problem for their edges. However, optimal solutions are not necessarily uniquely determined. (B) shows a contracted graph after the top-left edge with 'a' and 'b' is substituted with a new vertex labeled with 'ab'. On the other hand, (C) shows a contracted graph after the top-right edge with 'a' and 'b' is substituted. This example implies that GRAPHITOUR can generate different compressed graphs.

In order to measure the similarity of networks, the same compressed graph should

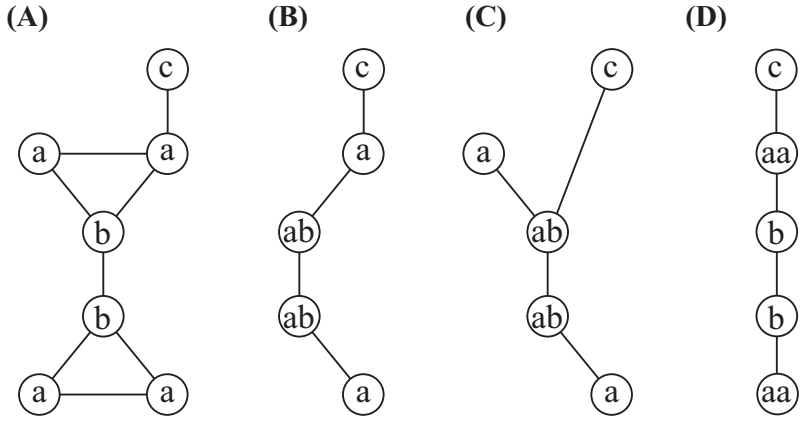

Figure 1: Example of contraction of identical edges. (A) Graph with 7 vertices and 8 edges. (B, C) graphs contracted from the graph (A) by GRAPHITOUR. (D) the graph contracted from the graph (A) by our proposed method.

always be obtained. Therefore, we improve GRAPHITOUR for that purpose, and propose the following algorithm to uniquely determine contracted edges in each iteration.

Procedure
Input: undirected graph $G(V,E)$ with labeled vertices V and edges E
(a total order $\leq$ is defined for the set of labels L,
and each $v \in V$ is labeled with $l_v \in L$);
Output: induced compression rules $\mathscr{R}$ and compressed graph;
Begin
$\mathscr{R} := \emptyset$;
$s(l) := \{l\}$ for each label $l \in L$;
while $|E| > 0$
$E(l_1,l_2) := \{(v_1,v_2)(\in E) | (l_{v_1},l_{v_2}) = (l_1,l_2)$ where $l_{v_1} \geq l_{v_2}, l_1 \geq l_2\}$;
$\mathscr{E} = \{E(l_1,l_2)$ for all $l_1,l_2 \in L |$ no two edges in $E(l_1,l_2)$ share a common vertex$\}$;
if $\mathscr{E} = \emptyset$ then **return** $(\mathscr{R},G)$;
$\mathscr{E}' := \{E(l_1,l_2)(\in \mathscr{E}) | \ |E(l_3,l_4)| \leq |E(l_1,l_2)|$ for all $E(l_3,l_4) \in \mathscr{E}\}$;
select $E(l_1,l_2)(\in \mathscr{E}')$ such that $s(l_1) \cup s(l_2) < s(l_3) \cup s(l_4)$
or $(s(l_1) \cup s(l_2) = s(l_3) \cup s(l_4)$ and $(l_1,l_2) < (l_3,l_4))$ for all $E(l_3,l_4) \in \mathscr{E}'$,
where $l_1 \geq l_2, l_3 \geq l_4$;
add a new label l_n in L such that $l_n > l$ for all $l \in L$;
$s(l_n) := s(l_1) \cup s(l_2)$;
$\mathscr{R} = \mathscr{R} \cup \{l_n \leftarrow (l_1,l_2)\}$;
for each edge $e \in E(l_1,l_2)$
substitute e with a new vertex labeled with l_n;
return $(\mathscr{R},G)$;
End

The proposed algorithm avoids to contract edges which share a common vertex. In the

example of Fig. 1, our algorithm does not choose edges whose endpoints are 'a' and 'b', instead chooses the second candidate edges whose endpoints are 'a' and 'a', and obtains the graph of (D) as the result. It should be noted that the proposed algorithm does not solve maximum cardinality matching problem because it selects only edges such that all edges with the same labels do not share a common vertex.

However, it is not sufficient to uniquely determine contracted edges because there can be more than one set which has the same number of edges, that is, $|\mathscr{E}'| > 1$. Therefore, we introduce a total order to sets of labels to determine priority of edges. Each edge has a set of labels (l_1, l_2) corresponding to two vertices of the edge. Let s_1 and s_2 be sets of labels. We can define a total order for s_1 and s_2 as follows. First, we sort s_1 and s_2 by descending order, respectively. We compare i-th elements $s_1^{(i)}, s_2^{(i)}$ of s_1 and s_2, and define $s_1 < s_2$ if i exists such that $s_1^{(i)} < s_2^{(i)}$ and $s_1^{(j)} = s_2^{(j)}$(for all $j < i$). The proposed algorithm selects edges with smallest set of labels from $\mathscr{E}'$ according to the total order. For example, if we compare $s_1 = \{l_1, l_3\}$ with $s_2 = \{l_3, l_2\}$ under $l_1 < l_2 < l_3$, s_1 and s_2 are sorted as $\{l_3, l_1\}$ and $\{l_3, l_2\}$, respectively, and we have $s_1 < s_2$.

When edges with (l_1, l_2) are contracted, a new label l_n is added to L, where $l_n > l$ for all $l(\neq l_n) \in L$. In computational experiments, Morgan index [11] based on graph structures is assigned to each vertex. However, new added labels themselves do not reflect the original graph structure. Therefore, in order to make effective use of the total order of original labels, we introduce a set of labels for each label l, $s(l)$, which consists of only original labels. Then, $s(l_n)$ is defined to be $s(l_1) \cup s(l_2)$ when (l_1, l_2) is substituted with l_n. The algorithm compares $s(l_1) \cup s(l_2)$ with $s(l_3) \cup s(l_4)$ before comparing edges of (l_1, l_2) and (l_3, l_4). For example, for the graph of Fig. 1D, the algorithm selects edges with ('aa', 'b') as contracted edges because it appears most frequently. However, if there is another edge with ('b', 'b') than shown in Fig. 1D, edges of ('aa', 'b') and ('b', 'b') are compared. We suppose that 'a'<'b'<'c'<'aa' and 'aa' was obtained by contracting edges with ('a', 'a'). Then, the corresponding sets to ('aa', 'b') and ('b', 'b'), $s_1 = s(\text{'aa'}) \cup s(\text{'b'}) = \{\text{'a','a','b'}\}$ and $s_2 = s(\text{'b'}) \cup s(\text{'b'}) = \{\text{'b','b'}\}$, are compared, sorted as $\{\text{'b','a','a'}\}$ and $\{\text{'b','b'}\}$, respectively, and we have $s_1 < s_2$. Then, edges with ('aa', 'b') are selected, and contracted to vertices with a new label 'aab', where 'aab'>'aa' and $s(\text{'aab'}) = s(\text{'aa'}) \cup s(\text{'b'}) = \{\text{'a','a','b'}\}$.

3 Similarity Measure

The universal similarity metric (USM) was proposed by Li *et al.* [8], and has been applied to several biological data [3, 7]. USM between two objects o_1 and o_2 is defined using Kolmogorov complexity $K(o)$ as follows:

$$USM(o_1, o_2) = \frac{\max(K(o_1|o_2^*), K(o_2|o_1^*))}{\max(K(o_1), K(o_2))}. \tag{1}$$

Kolmogorov complexity $K(o)$ of an object o is defined to be the length of the shortest program P for a universal Turing machine U which outputs o, and the conditional Kolmogorov complexity of o_1 given o_2 is defined to be the length of the shortest program P which outputs o_1 when o_2 is given as follows:

$$\begin{cases} K(o) = \min\{|P| \mid P \text{ is a program such that } U(P) = o\}, \\ K(o_1|o_2) = \min\{|P| \mid P \text{ is a program such that } U(P, o_2) = o_1\}. \end{cases} \tag{2}$$

Table 1: Statistics of metabolic pathways for several organisms.

organism	# of nodes	# of edges
H. sapiens	1550	1673
M. musculus	1518	1640
A. thaliana	1389	1395
D. melanogaster	1238	1250
C. elegans	1049	1009
E. coli	1103	1256
S. cerevisiae	983	1028
B. subtilis	994	1065

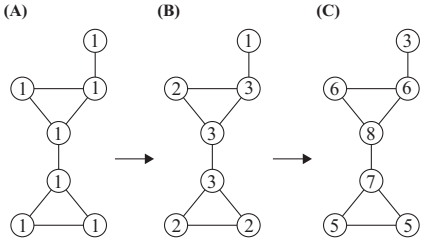

Figure 2: Example of calculation of Morgan index. (A) 1 is assigned to each vertex. (B) first iteration. (C) second iteration.

It should be noted that $K(o)$ is considered as a measure of the amount of information which the object o contains.

Since it is not possible to obtain these Kolmogorov complexities for real data, we approximate $K(G)$ of a graph G by $C(G) = |\mathscr{R}| + |E_c|$, where $|\mathscr{R}|$ means the number of rules extracted from G by our method, and $|E_c|$ means the number of remaining edges after the compression of G. The conditional Kolmogorov complexity $K(G_1|G_2)$ of G_1 given G_2 can be approximated to be $C(G_1 \cup G_2) - C(G_2)$ as in [3, 7], where $G_1 \cup G_2$ means the concatenated graph $G'(V', E')$ of $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ such that $V' = V_1 \cup V_2$, $E' = E_1 \cup E_2$, $|V'| = |V_1| + |V_2|$ and $|E'| = |E_1| + |E_2|$. Even if there are identical vertices (i.e. vertices with identical labels) between G_1 and G_2, they are added to V' as different vertices.

Substituting $K(o)$ of Eq.(1) with $C(G)$, the approximated USM for graph compression, GUSM, between two graphs G_1 and G_2 is given as follows:

$$GUSM(G_1, G_2) = \frac{C(G_1 \cup G_2) - \min(C(G_1), C(G_2))}{\max(C(G_1), C(G_2))}. \tag{3}$$

It should be noted that $GUSM(G, G) = 0$ if $|E_c| = 0$. If G_1 and G_2 are similar, $GUSM(G_1, G_2)$ approaches 0.

4 Computational Experiments

To evaluate the proposed measure, we used metabolic pathways for several organisms, *H. sapiens*, *M. musculus*, *A. thaliana*, *D. melanogaster*, *C. elegans*, *E. coli*, *S. cerevisiae*, and *B. subtilis*, from KEGG database [4] (see Table 1).

In our first computational experiment, all nodes in the metabolic networks were labeled with chemical compounds, and there was only one edge having the same labels, that is, $|E(l_1, l_2)| = 1$. Then, our compression algorithm for $G(V, E)$ produced rules $\mathscr{R}$ and the remaining graph $G_c(V_c, E_c)$ as $C(G) = |\mathscr{R}| + |E_c| = |E|$. This means that G is not compressed.

Since we would like to compare network structures for the metabolic networks, we replaced labels with Morgan index [11]. Fig. 2 shows an example of calculation of Morgan index. First, 1 is assigned to each node. Next, the sum of values of adjacent nodes

is assigned for each node. This iteration is repeated until the number of different values of Morgan index does not increase. We call the index obtained in this way the original Morgan index. Morgan index of one iteration is equivalent to the degree of each node, and Morgan index depends on graph structures.

We fixed the number of iterations of the Morgan index procedure, applied our compression algorithm to individual and concatenated metabolic networks, $G_1, G_2, G_1 \cup G_2$, and calculated $GUSM(G_1, G_2)$ from $C(G_1)$, $C(G_2)$ and $C(G_1 \cup G_2)$. To confirm that our compression algorithm works for measuring the similarity of metabolic networks, we obtained hierarchical clustering results using the nearest neighbor (single linkage) method, and compared them with actual phylogenetic trees. Moreover, we performed such experiments with several numbers of iterations from 1 to 20 because the number of iterations of the original Morgan index is at most 11 for the metabolic networks.

Fig. 3 shows the results of hierarchical clustering for metabolic networks of several organisms, *H. sapiens*, *M. musculus*, *A. thaliana*, *D. melanogaster*, *C. elegans*, *E. coli*, *S. cerevisiae*, and *B. subtilis* with Morgan indices of 1, 2, 3, 6, 11, and 12 iterations. The numbers of contracted edges for the metabolic network of *H. sapiens* with Morgan indices of 1, 2, 3, 6, 11, and 12 iterations were 251, 1367, 1387, 1395, 1395, and 1395, respectively. The results of more than 5 iterations were almost similar to those of 12 iterations. Fig. 4 shows the results on the number of different values of Morgan indices for the metabolic networks for 1-20 iterations of the Morgan index procedure. We can see from this figure that the number of different values of Morgan indices is almost constant for more than 11 iterations. For a small number of iterations, it is considered that metabolic networks were not compressed well because many edges have the same labels and share common nodes. This means that the number of iterations is required to be large for measuring the similarity more accurately. However, for that purpose, the maximum number of iterations of the original Morgan index over all organisms is sufficient because the number of different values of Morgan indices is almost constant in more than that.

According to the results of hierarchical clustering in Fig. 3, *H. sapiens* was always nearest to *M. musculus* among the metabolic networks. Bacterial organisms of *B. subtilis* and *E. coli* were furthest from *H. sapiens* in the result of 12 iterations. It is considered that the result of 12 iterations is almost consistent to actual phylogenetic trees. This suggests that the proposed method could measure the similarities between metabolic networks adequately.

Furthermore, the proposed method is efficient. The computational time was at most 9 seconds even for the concatenated network of *H. sapiens* and *M. musculus* with Morgan index of 12 iterations. These experiments were done in a single processor core on a PC with Xeon X5460 3.16GHz CPUs and 8GB memory under the Linux (version 2.6) operating system, where the g++ compiler was used with optimization option -O3.

5 Concluding Remarks

In this paper, we have proposed a novel method for compressing biological networks. One of the important properties of the proposed method is that isomorphic networks are compressed in the same way. We have applied the proposed compress method to comparison of metabolic networks. The results suggest that the proposed compression method is useful for comparison of biological networks although comparison with exiting methods

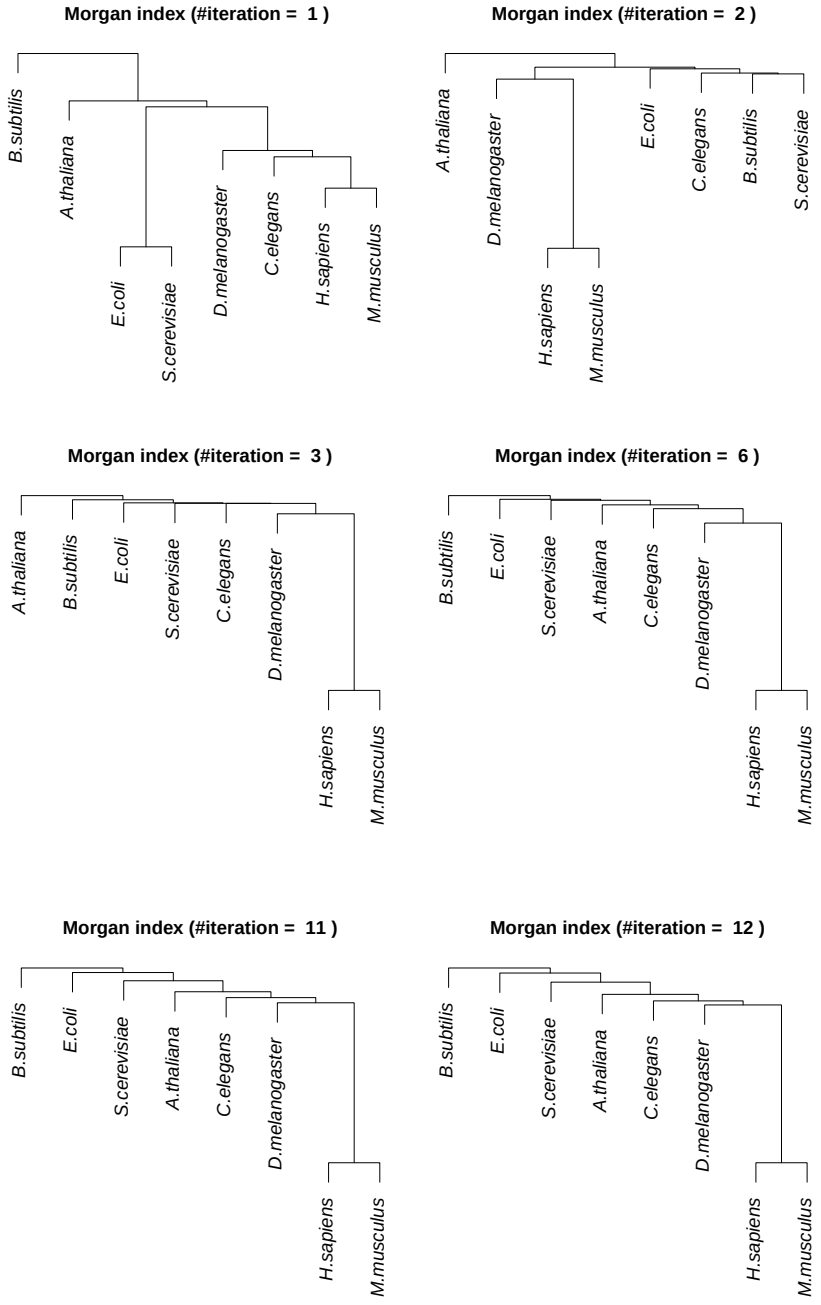

Figure 3: Results of hierarchical clustering for metabolic networks of several organisms, *H. sapiens*, *M. musculus*, *A. thaliana*, *D. melanogaster*, *C. elegans*, *E. coli*, *S. cerevisiae*, and *B. subtilis* with Morgan indices of 1, 2, 3, 6, 11, and 12 iterations.

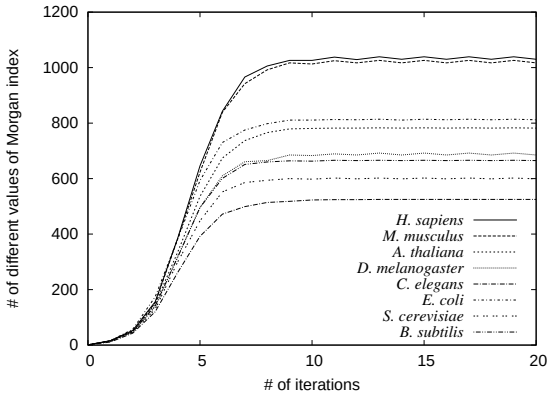

Figure 4: Results on the number of different values of Morgan indices for metabolic networks of several organisms, *H. sapiens*, *M. musculus*, *A. thaliana*, *D. melanogaster*, *C. elegans*, *E. coli*, *S. cerevisiae*, and *B. subtilis* for 1-20 iterations of the Morgan index procedure.

is left as future work.

Though we have applied the method to comparison of networks, the application is not limited to comparison. It might be applied to detection of network motifs with hierarchical structures because our method iteratively compresses edges (edges can be replaced by small subgraphs).

One drawback of our proposed compression method is that it is not a lossless compression method (i.e., the original network cannot be reconstructed from compressed data). Therefore, improvement of the method towards lossless compression is also important future work.

Acknowledges

This work was partially supported by Grants-in-Aid #19650053 and #21700323 from MEXT, Japan.

References

[1] Adler, M., Mitzenmacher, M.: Towards compressing web graphs, *2001 Data Compression Conference*, 203-212, 2001.

[2] Cook, D. J., Holder, L. B.: Substructure discovery using minimum description length and background knowledge, *Journal of Artificial Intelligence Research*, 1:231-255, 1994.

[3] Hayashida, M., Akutsu, T.: Image compression-based approach to measuring the similarity of protein structures, *Proc. 6th Asia-Pacific Bioinformatics Conference*, 221-230, 2008.

[4] Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., Yamanishi, Y.: KEGG for linking genomes to life and the environment, *Nucleic Acids Research*, 36:D480-D484, 2008.

[5] Kelley, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R., Ideker, T.: PathBLAST: a tool for alignment of protein interaction networks, *Nucleic Acids Research*, 32:W83-W8, 2004.

[6] Kocsor, A., Kertész-Farkas, A., Kaiàn, L., Pongor, S.: Application of compression-based distance measures to protein sequence classification: a methodological study, *Bioinformatics*, 22:407-412, 2005.

[7] Krasnogor, N., D. A. Pelta, D. A.: Measuring the similarity of protein structures by means of the universal similarity metric, *Bioinformatics*, 20:1015-1021, 2004.

[8] Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P., Zhang, H.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, 17:149-154, 2001.

[9] Li, Z., Zhang, S., Wang, Y., Zhang, X.-S., Chen, L.: Alignment of molecular networks by integer quadratic programming, *Bioinformatics*, 23:1631-1639, 2007.

[10] Liang, Z., Xu, M., Teng, M., Niu, L.: NetAlign: a web-based tool for comparison of protein interaction networks, *Bioinformatics*, 22:2175-2177, 2006.

[11] Morgan, H.: The generation of unique machine description for chemical structures - a technique developped at chemical abstracts service, *Journal of Chemical Documentation*, 107-113, 1965.

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

[13] Peshkin, L.: Structure induction by lossless graph compression, *Proc. 2007 Data Compression Conference*, 53-62, 2007.

[14] Pinter, R. Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, N.: Alignment of metabolic pathways, *Bioinformatics*, 21:3401-3408, 2005.

[15] Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection, *Proc. Natl. Acad. Sci. USA*, 105:12763-12768, 2008.

[16] Tohsato, Y., Matsuda, H., Hashimoto, A.: A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy, *Proc. 8th Int. Conf. Intelligent Systems for Molecular Biology*, 376-383, 2000.

[17] Wernicke, S., Rasche, F.: Simple and fast alignment of metabolic pathways by exploiting local diversity, *Bioinformatics*, 23:1978-1985, 2007.

[18] Yang, J., Savari, S. A., Mencer, O.: An approach to graph and netlist compression, *Proc. 2008 Data Compression Conference*, 33-42, 2007.

[19] Zaslavskiy, M., Bach, F., Vert, J-P.: Global alignment of protein-protein interaction networks by graph matching methods, arXive:Math:0905.1106v1.