GOMA: Web Utility for Direct Finding of Enriched Gene Ontology Terms from Gene Expression Profile

Eriko Mizutani and Jun Sese

Abstract— **Microarrays have become popular devices used to elucidate changes of cell status. Following microarray experiments, technicians would like to verify their observations and changes of cell status occurring because of stimuli during experiments. Currently, we validate analyses of several or hundreds of samples using clustering and visualization. However, comparison is not an easy task because knowledge of data analysis is required and hundreds of clusters might be produced.**

This study presents GOMA, a web-based expression enrichment checking server using Gene Ontology (GO). By putting sets of gene names and expressions into the server, which GO categories' genes are enriched can be verified immediately, and changes of cell status can be detected. The server presents a graphical GO structure association of their terms to ease the checking scheme. We evaluate that our web server enables us to recognize functions associated with changes of gene expression and to elucidate effects of different stimuli. The experimental results shows that our web server detects aldo-keto reductase activity under sorbitol supplemental condition from yeast expression data automatically, and our server visualizes the functional changes between sorbitol condition and stationary phase. GOMA is available online at http://goma.sel.is.ocha.ac.jp/ .

I. Introduction

Microarrays have become widely used to observe transcriptional changes. Recently, new sequencing techniques also give us a gene expression profile. A common analytical flow of the expression data is first clustering and then associating clusters with GO terms or pathways. Web sites and tools for analyses of microarray data such as DAVID [1] and GO::TermFinder [2] have been proposed to ease the search for relations among them.

In the microarray test, because most of the experimental process is done by machine, experimenters would like to verify their experiments and changes of gene expressions by stimulus. However, it is not easy because skill at finding clusters associating GO terms differ from doing experiments. Most technicians and experimenters are not familiar with analyses of large amounts of data.

To assist experimenters in checking gene expression changes promptly, we propose a web server called Gene Ontology Manipulating Array results server (GOMA). [1] Fig. 1 presents an overview of the GOMA system. GOMA accepts one sample containing thousands of genes' expressions. Data can be input from a web browser. GOMA holds Gene Ontology data on a MySQL database server. Its data are distributed by the Gene Ontology Consortium [3]. Then GOMA extracts statistically significant GO terms from the inputs; it shows the top N significant terms and their p-values. The terms might have direct parent-child connections. Therefore, GOMA displays relations of the GO terms. It is difficult and verbose to show all the terms because about 25,000 GO terms exist. For that reason, GOMA selects around the top N terms. The results have links to a Gene Ontology site.

The problems of constructing this site are:

1) extraction of significant terms, and
2) visualization of the terms.

We rank the significant terms using the Mann-Whitney U, a widely used statistical index. The computations are independent from the method to observe gene expression. We introduce an accurate calculation of the index, which allow us to compute p-value over 25,000 GO terms with about 6,000 genes within 20 seconds. We also present that the index can elucidate effects of different stimuli. GO forms a directed acyclic graph (DAG) structure, which is a graph having no cycle. Generic drawing method might arrange the the top N GO terms as mutually distant. Furthermore, computation of the arrangement having minimum number of crossings is NP-hard problem. We show the DAG graph of selected GO terms with the restriction that the top N terms locate in the middle. We propose a graph arrangement method that is suitable for a web server.

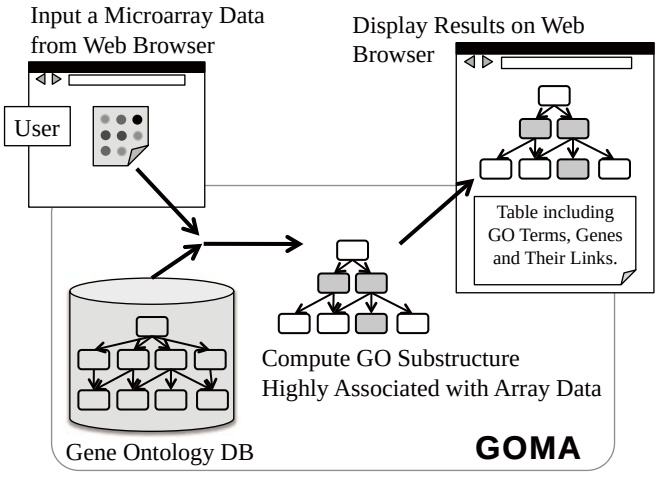

Fig. 1. Overview of the GOMA Web Server

This work was partially supported by Grant-in-Aid for Young Scientists (B) (20700268) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

The authors are with Department of Computer Science, Ochanomizu University, 2-1-1 Otsuka, Bunkyo, Tokyo, 112-8610, Japan. {mizutani,sesejun}@sel.is.ocha.ac.jp

[1] *GOMA* is a Japanese word meaning sesame.

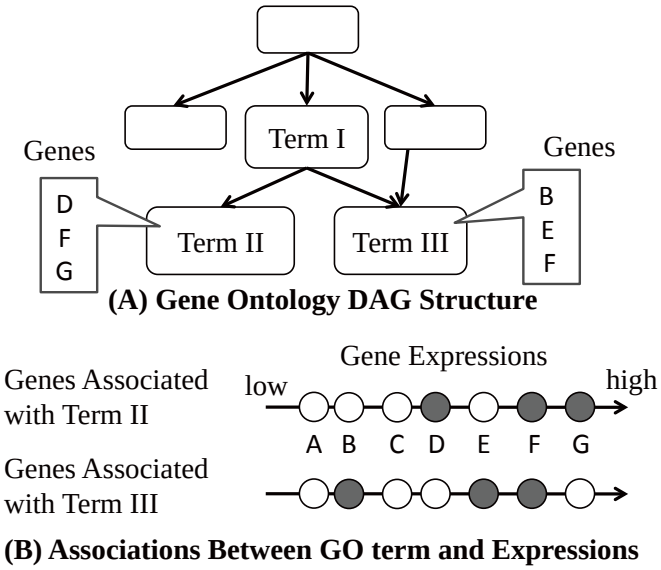

Fig. 2. Division of Gene Expressions by GO terms

Section II describes related work. In Section III, we introduce the efficient ranking of statistically significant terms. Section IV describes visualization of enriched terms with a DAG structure. We demonstrate our GOMA web site and demonstrate the efficiency of our statistical model and visualization. We conclude this report in Section VI.

II. Related Work

Various tools [4] to use GO have been proposed. A useful application of GO is to interpret microarray data. For that purpose, several web sites and programs such as DAVID [1] and GO::TermFinder [2] have been proposed. However, all such programs require a set of genes. The set is often generated by clustering of gene expressions. In contrast, our web site can be used without cluster analysis. For single sample analysis, gene set enrichment analysis has been proposed [5]. This method requires much computational time to compute a rigorous p-value of a subset of genes. It is unsuitable for use as a web-based tool.

DAVID and servers using GO::TermFinder show us the Gene Ontology structure of significantly associated terms. Actually, DAVID describes GO as a tree structure. GO forms a DAG, which might have many parents of each term; therefore duplicate branches might be described. Some methods described in previous reports such as GO::TermFinder [2] and AmiGO [3] draw a DAG structure to avoid the problem. The software programs use a generic graph layout method. Therefore, they might generate a confusing structure to interpret GO terms. For example, significantly associated terms with the user input genes might be shown as mutually distant. Therefore, GOMA arranges GO terms under the restriction that the positions of statistically significant terms be central in a graph.

III. Ranking Enrichment Terms

In this section, we rank statistically significant Gene Ontology (GO) terms from the data, and show the enriched terms from user input gene expression values as observed using microarrays.

Fig. 2 shows an example of selection of enriched GO terms. In Fig. 2(A), each rounded square and edge respectively signify a GO term and a parent-child relationship between terms. Every GO term is associated with genes; term II is associated with genes D, F, and G, whereas term III is related to genes B, E, and F. Genes associated with term T are also associated with ascendants of T. Therefore, term I is connected to genes B, D, E, F, and G, which is a union of genes associated with terms II and III. Fig. 2(B) represents associations between a term and expressions. Each circle denotes expression values. Term II is associated with genes D, F, and G; the genes are shown as gray.

In the comparison of terms II and III, term II is more significantly related than term III in the experiment because all genes associated with term II have high expression values, although genes associated with term III include both high and low expression genes. Representation by quantities of the significance enable us to compute the statistical significance of terms and to rank terms according to the values.

Various indexes to measure enrichment of a set of genes have been proposed. Most indexes require much computing time, and are therefore unsuitable for use on a web site. The web site requires computation in constant time and returns results as soon as possible. We here use Mann-Whitney U (Wilcoxon's rank sum test), which is a well known statistical measure. The measure depends only on ranks of values. For that reason, the computations are independent from the method to observe gene expression. We introduce statistical index U to find GO term enrichment; we also propose a technique to compute the value quickly to use it on the web site. We show that the U is sufficiently sensitive to detect whether the user's expression has significant changes through the case study in Section V.

The formula for Mann-Whitney U is:

1) Arrange all expressions which are user-input and associated with GO into a single ranked series.
2) Add up the ranks for the genes associated with a GO term. We define the genes and their sum respectively as G and r.
3) $U = r - |G| \times (|G| + 1)/2$

From this U, we can compute p-value.

GO contains about 25,000 terms. We must compute this U for all terms. In each calculation, the most time-consuming part is ordering expressions into a single ranked series. However, the orders over all the GO terms are the same because the total gene set used in the computation is the same. This observation enables us to reduce the computing time.

For example, Fig. 2(B) contains gene expressions of two different terms. All user input gene expressions are sorted. The lowest gene is "A", whereas the highest is "G". We add

numbers to genes according to their rank: "A" is first and "G" is seventh. The order is the same between terms II and III. For that reason, we must sort the expressions only once. For term II, associated genes are ranked 4th, 6th, and 7th. Then, $U = 4+6+7-3\times 4/2 = 11$ and its p-value is 0.114. For term III, $U = 7$ and its p-value is 0.857. Therefore, term II is more significant than term III. These calculations enable us to rank GO terms by sorting the data once.

We choose GO terms that have the N lowest p-values. Here, N is a user specified value. In fact, GOMA shows the terms in the graph layout as well as those in the HTML table. In the next section, we discuss the graph visualization of the GO terms.

IV. Visualization of Enriched Terms

Graph visualization of the top (lowest) N GO terms makes it easy to understand changes that occur within cells. The GO term viewers in DAVID [1] and GO::TermFinder [2] use tree structure visualization by HTML and GO's DAG structure using generic graph drawing method, respectively. Nodes having more than one parent appear more than once in a tree when drawing a DAG structure as a tree. Generic graph drawing method and GraphViz [6], which is widely used graph visualization software, might arrange the top N GO terms as mutually distant. Furthermore, when the number of terms to display is large, it is difficult to identify the parent-child relations.

These software programs subsume that the input genes are mutually related very closely; therefore, the visualizing terms are close to each other on the DAG structure. On the other hand, significantly changed top N terms might contain several different biological functions. This possibility prompts us to devise a method for visualizing GO terms even if the distance between the GO terms on DAG is great.

We herein introduce a visualization method that retains parent-child term relations and enables us to identify the top N terms easily.

A. Term Selection and Layered Assignment

We first select terms to display and then draw a layered layout [7] that requires the following two steps: (1) Layered assignment and (2) Crossing reduction.

The top N terms might be located in various branches in GO's DAG structure. We first find a common ascendant a of the top N terms to display all the relations between the top N terms. We then extract all terms on all paths between the a and top N terms. We define T_D as all displayed terms:

$$T_D = \{t' \mid \text{term on path from } a \text{ to } t \in T\}.$$

We next assign a layer of $t \in T_D$ for a layered drawing. Let L_d be a set of terms in d-th layer. We assign the layer of term t according to the longest path from a to t to avoid the up-arrow. For example, if the length of the longest path from a to t is three, t is assigned to layer L_3.

B. Crossing Reduction

In spite of the use of layered assignments, the positions of the top N terms might be mutually distant, or it might be difficult to understand parent-child relations by crossings of lines between terms. We therefore reduce crossings under the restriction that all top N terms locate in the middle of the graph.

We first arrange terms at regular intervals in each layer. We divide the draw width into $n+1$ equal parts and locate terms between the neighbor parts when layer L_d contains n terms. Both sides are margins. Fig. 3(A) portrays an example of initial positions. Rounded squares and arrows respectively denote a term and a parent-child relation. In this figure, seven terms are described as $t_1, \ldots, t_9$. L_1 contains $\{t_2, t_3\}$. Gray rounded squares denote the top N terms.

Let t be a term. We define $children(t)$ and $parents(t)$ as children and parent terms of t, respectively. For example, in Fig. 3(A), $children(t_3) = \{t_6, t_7\}$. Let us define the term position $x(t)$. Assume that term t exists in L_d, we assign a count from the left to $x(t)$. In the running example, $x(t_2) = 0$ and $x(t_6) = 2$. We define

$$avg(T) = \frac{1}{|T|}\sum_{t\in T} x(t),$$

where T is a set of terms. In addition, $avg(T)$ calculates the average position of terms in T.

Let us move the top N terms in the middle of each layer. We move the top N term in layer L_d using the following procedure. Let T_N be the set of top N terms.

1) Select term $t \in T_N \cap L_d$.
2) Select term $t' \in L_d$ randomly.
3) If $|x(t') - avg(L_d)| < |x(t) - avg(L_d)|$, exchange the position between t and t'.

For example, let us move t_4 in Fig. 3(A) belonging to the top N terms and locating the left end in L_2. Presume that

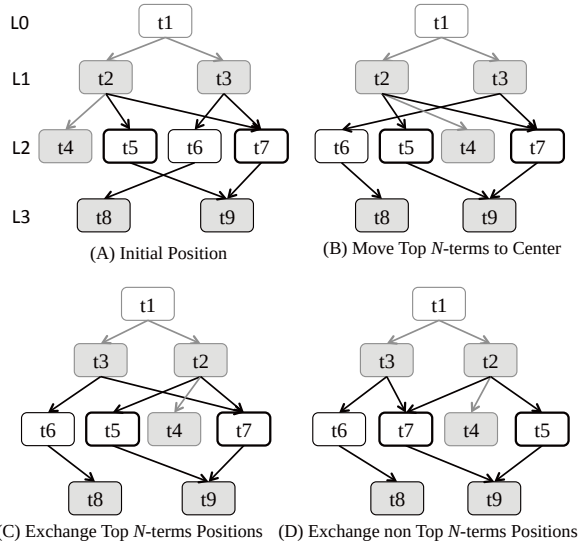

Fig. 3. Ordering Terms for Visualization of Top N-terms. Gray boxes denote the top N-terms.

we select t_6 from L_2. Here, t_6 is closer to the middle than t_4; therefore, we exchange t_4 with t_6. Fig. 3(B) shows the status after moving. We perform this procedure 100 times for each layer.

We next reduce the number of crosses. Although various crossing reduction methods have been proposed, most methods might take much time because the problem to compute its optimal solution is NP-hard [7]. Most of GO's DAG structure is not complicated; that is, only some terms have multiple parents. Furthermore, we have two restrictions. We must compute in constant time to show the result on a web browser. Moreover, the top N terms are fixed close to the middle. Most methods make it difficult to handle the restrictions.

We scan from the root to leaves four times to arrange positions of terms under the restrictions. During the first two times, we exchange the positions of the top N terms; during the next two times, we exchange the positions of terms excepting the top N terms.

In first and third scannings, we reduce the number of edge crossings between each term and its children terms.

1) select two terms t_1 and t_2 ($x(t_1) < x(t_2)$) randomly from $L_d \cap T_N$ (at the first sweep), or $L_d - T_N$ (at the third sweep).
2) if t_1 or t_2 has no children, go to step 1 and re-select new terms.
3) if $avg(children(t_1)) > avg(children(t_2))$, then exchange positions between t_1 and t_2.

Let d be the height of the DAG to display. We do this procedure from L_0 to L_{d-2}. We repeat this selection 100 times for each layer. In this procedure, we check the average positions of terms and exchange their positions if the order of terms is not equal to the average position order of their children.

For example, in Fig. 3(B), we try to exchange the position of t_2 with t_3. $children(t_2) = \{t_4, t_5, t_7\}$ and $children(t_3) = \{t_6, t_7\}$. Then, $avg(children(t_2)) = 2$ and $avg(children(t_3)) = 1.5$. From this result, $avg(children(t_2)) > avg(children(t_3))$, while $x(t_2) < x(t_3)$. Therefore, we exchange the position of t_2 with t_3 (Fig. 3(C)).

In the second and fourth scannings, we reduce the number of edge crossings between each term and its parent terms. We do this procedure from L_1 to L_{d-1}, and repeat the selection 100 times for each layer.

1) select two terms t_1 and t_2 ($x(t_1) < x(t_2)$) randomly from $L_d \cap T_N$ (at the second sweep), or $L_d - T_N$ (at the fourth sweep).
2) if $avg(parents(t_1)) > avg(parents(t_2))$, exchange positions between t_1 and t_2.

Presume that t_5 and t_7 are selected as exchange candidates in Fig. 3(C). $parents(t_5) = \{t_2\}$ and $parents(t_7) = \{t_2, t_3\}$. $avg(parents(t_5)) = 1$ and $avg(parents(t_7)) = 0.5$; then we exchange the position of t_5 with t_7.

Using these four sweeps, we reduce the number of crossings under the restriction that the top N terms are close to the middle and that the number of exchanges of nods is constant.

In the visualization on the web, we show two different views: a thumbnail view and a whole view. The thumbnail view shown in Fig. 4 represents overview of the DAG structure. Using the thumbnail view, we can recognize the changes of statistically significant terms without checking details of the terms. The whole view displayed in Figs. 5 and 6 shows us the complete DAG structure and term names associated with the top N terms. We describe three different DAG structures according to three categories of GO.

V. Case Study

In this section, we show the usefulness of our GOMA for checking experimental results. In our site, a user can put expression data into an input area on the GOMA website. The inputs are expected to contain sets of gene and expression values. After input, the data and selection of your species, only a single click of the submit button is necessary. In the test described below, we use the gene expression profile of yeast [8] and input 6,152 genes with their expressions under one condition.

A. Sorbitol Supplemental Condition

Presume that we have observed expressions of yeasts grown after 2h under a sorbitol supplemented condition. After inputting the gene expression profile and pressing the submit button, DAG structures are visible in Fig. 4, consisting of three different GO categories.

Three graphs shown in Fig. 4 are associated respectively with GO term categories: "Molecular Function", "Biological Process" and "Cellular Component". Each rounded square shows the term and the orange rectangle shows the top 10 terms. All top 10 terms are located close to the middle in this figure.

From the left-most graph representing Molecular Function, we recognize that top 10 terms are divided into two branches. To see its details, by clicking the figure, you will see Fig. 5 with a table including details of GO terms (HTML version of Table I). Duplicate branches would be described for the left branch if we describe this graph in a tree structure. The width of the figure might be larger if we draw the same DAG using generic method.

Table I shows details of the top 10 terms in biological processes. This table contains ranks of terms, GO term ids, the term names, p-values calculated using the Mann-Whitney test, and quantities of user input genes associated with the terms. The input expression profile is under a sorbitol condition. Yeast converts sugars into cellular energy and thereby produces ethanol. In this process, aldo-keto reductions are required. In this table, aldo-keto reductase activity is ranked 4th. This result shows to the experimenters that this observation would be successful.

To verify whether our index, Mann-Whitney U, is sensitive for re-experimentation, we compute our index from expression values of yeast under sorbitol conditions after 45 min. The results are depicted in Table II. In the comparison with terms enriched in 2h, aldo-keto reductase activity is also ranked 6th. This verifies to us that important cell changes

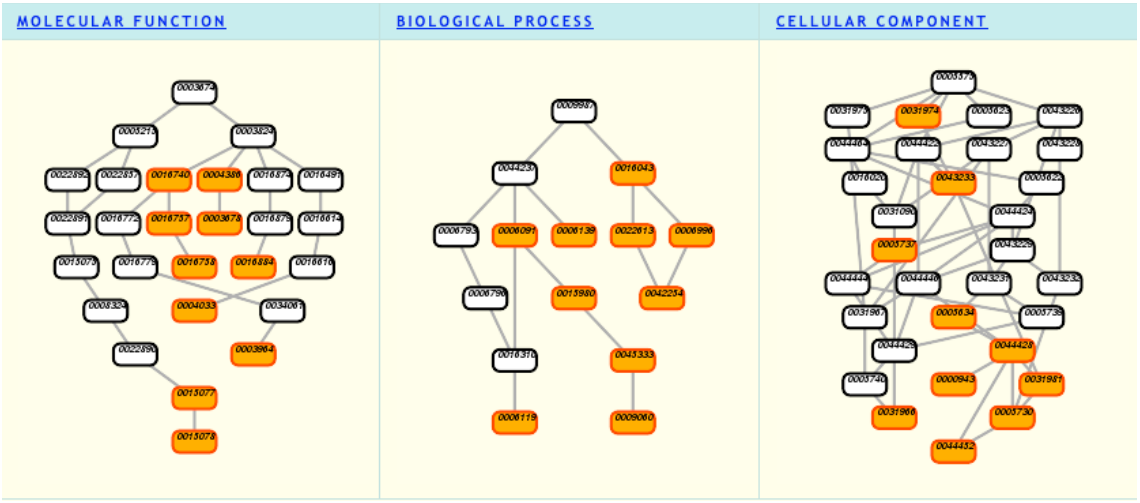

Fig. 4. Thumbnail view for yeast grown after 2h under a sorbitol supplemental condition

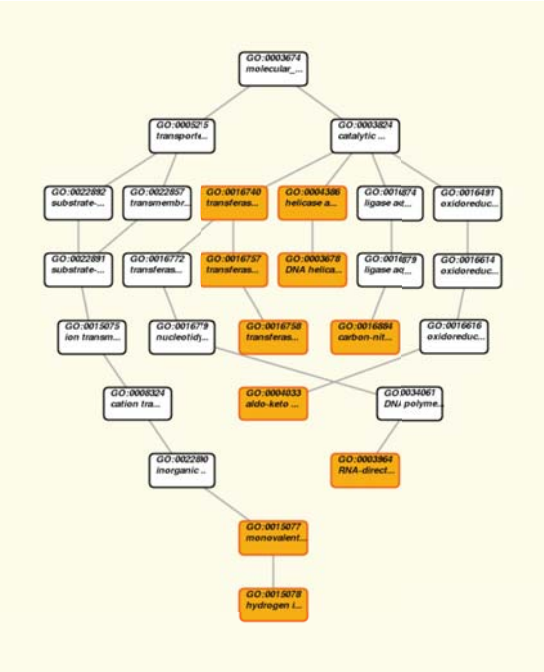

Fig. 5. Graph of top 10 molecular function terms with yeast expressions grown after 2h under sorbitol supplemental conditions (Details are in Table I)

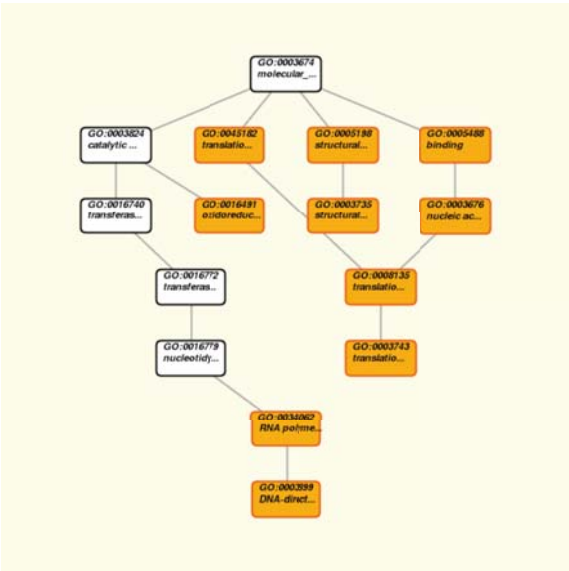

Fig. 6. Graph of top 10 molecular function terms with yeast expressions observed at the stationary phase

TABLE I

TOP 10 MOLECULAR FUNCTION TERMS WITH YEAST EXPRESSION GROWN AFTER 2H UNDER SORBITOL SUPPLEMENTAL CONDITIONS

Rank	GO ID	Term Name	*p*-value	Num of Genes
1	GO:0004386	helicase activity	9.56e-07	69
2	GO:0016740	transferase activity	2.92e-06	520
3	GO:0016757	transferase activity, transferring glycosyl groups	1.24e-04	81
4	GO:0004033	aldo-keto reductase activity	1.74e-04	8
5	GO:0003964	RNA-directed DNA polymerase activity	2.10e-04	21
6	GO:0015077	monovalent inorganic cation transmembrane transporter activity	5.59e-04	48
7	GO:0003678	DNA helicase activity	5.75e-04	25
8	GO:0016758	transferase activity, transferring hexosyl groups	8.09e-04	64
9	GO:0015078	hydrogen ion transmembrane transporter activity	1.32e-03	43
10	GO:0016884	carbon-nitrogen ligase activity, with glutamine as amido-N-donor	1.54e-03	8

TABLE II

TOP 10 TERMS GROWN AFTER 45 MIN UNDER SORBITOL SUPPLEMENTAL CONDITIONS

Rank	GO ID	Term Name	p-value	Num of Genes
1	GO:0003676	nucleic acid binding	4.56e-10	407
2	GO:0008233	peptidase activity	6.12e-09	108
3	GO:0004386	helicase activity	6.75e-08	69
4	GO:0003677	DNA binding	1.86e-06	206
5	GO:0005488	binding	2.40e-06	842
6	GO:0004033	aldo-keto reductase activity	2.55e-06	8
7	GO:0003964	RNA-directed DNA polymerase activity	6.49e-06	21
8	GO:0034062	RNA polymerase activity	8.93e-06	32
9	GO:0003899	DNA-directed RNA polymerase activity	8.93e-06	32
10	GO:0016491	oxidoreductase activity	1.08e-05	215

TABLE III

TOP 10 TERMS OF THE STATIONARY PHASE

Rank	GO ID	Term Name	p-value	Num of Genes
1	GO:0003735	structural constituent of ribosome	3.22e-54	187
2	GO:0005198	structural molecule activity	4.95e-47	288
3	GO:0016491	oxidoreductase activity	3.81e-22	215
4	GO:0005488	binding	1.96e-15	842
5	GO:0003676	nucleic acid binding	9.20e-14	407
6	GO:0008135	translation factor activity, nucleic acid binding	2.48e-13	42
7	GO:0003743	translation initiation factor activity	4.09e-12	28
8	GO:0045182	translation regulator activity	3.43e-11	50
9	GO:0034062	RNA polymerase activity	5.34e-11	32
10	GO:0003899	DNA-directed RNA polymerase activity	5.34e-11	32

according to growth conditions would be detected using our method. On the other hand, gene regulations are activated in the 45 min sample, whereas transmembrane activities are repressed. This might be caused by differences of populations of cell cycle periods.

B. Stationary Phase

Next, we demonstrate GOMA with different gene expressions. We input expression values of yeast collected at time zero status after 8 h growth.

Fig. 6 shows a GO graph of the top 10 terms for the stationary phase. Comparison of the two figures, Fig. 5 and Fig. 6, shows that the two graphs present different structures. Especially, Fig. 5 depicts two children at the root node, whereas Fig. 6 portrays four children.

The top 10 terms are described in Table III. No common terms exist between Table I and Table III. This result verifies to us that top 10 terms affect cell status changes.

VI. CONCLUSION

Herein, we proposed a web server that enables us to check functional changes quickly from expression profiles. Although most web servers computing enrichment of Gene Ontology require a set of genes calculated by clustering analysis, most technicians and experimenters are unfamiliar for the statistical task. To overcome the problem, we introduce a new web server, GOMA, which requires only a set of gene names and expression profiles with no statistical analysis. GOMA detect the changes of molecular process of sorbitol supplemental condition automatically. Therefore, we believe GOMA to be an important contribution in the annotation pipeline.

Our future work is to add more user-friendly features to this web site. One is ready comparison of the GO tree and tables among microarrays. Currently, GOMA accepts only one sample. We would like to extend this feature to multiple samples. Another task is enrichment of the links to associate other databases such as KEGG [9] and reactome.org [10].

REFERENCES

[1] G. Dennis, B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki, "David: Database for annotation, visualization, and integrated discovery," *Genome Biology*, vol. 4, p. R60, 2003.

[2] E. Boyle, S. Weng, J. Gollub *et al*., "Go::termfinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes," *Bioinformatics*, vol. 20, pp. 3710–3715, 2004.

[3] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler *et al*., "Gene ontology: tool for the unification of biology. the gene ontology consortium." *Nat Genet*, vol. 25, no. 1, pp. 25–29, May 2000.

[4] P. Khatri and S. Drăghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems." vol. 21, pp. 3587–95, 2005.

[5] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proc. Natl. Acad. Sci.*, vol. 102, pp. 15 545–15 550, 2005.

[6] "Graphviz: http://www.graphviz.org/."

[7] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: algorithms for the visualization of graphs*. Prentice Hall, 1999.

[8] A. P. Gasch *et al*., "Genomic expression programs in the response of yeast cells to environmental changes," *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, 2000.

[9] M. Kanehisa and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[10] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein, "Reactome: a knowledgebase of biological pathways." *Nucleic Acids Res*, vol. 33, no. Database issue, January 2005.